Predicting Water Quality Variables

by

Reem Elmahdi

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

Supervisor: Prof. Willie Brink
Co-supervisor: Dr Josefine Wilms

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2020

Abstract

Water is an important substance for all of life, and can be used in domestic, agricultural and industrial activities. Water quality determines the usefulness of water for particular purposes, and can be defined in terms of time-varying water quality variables such as dissolved oxygen, turbidity, temperature, pH, specific conductance, chlorophylls, nitrate and salinity. Different mathematical and statistical models have been used for the prediction of time-series data. Machine learning can also be used when enough data is available. In particular, artificial neural networks (ANNs) have demonstrated success in solving such problems. They are conceptually simple and easily implemented. In this thesis, an overview of two ANN structures is presented for solving the problem of predicting water quality variables. Specifically, multilayer perceptrons (MLPs) and long short-term memory (LSTM) networks are presented. Experiments are conducted on Hog Island water quality variables and the results of the models are compared using various accuracy metrics like root mean squared error. It is found that LSTM performs better than MLP across most of the accuracy metrics.

Opsomming

Water is belangrik vir alle vorme van lewe, en kan in huishoudelike, landbou- en nywerheidsaktiwiteite gebruik word. Waterkwaliteit bepaal die bruikbaarheid van water vir spesifieke doeleindes, en kan gedefinieer word in terme van tydafhanklike waterkwaliteitsveranderlikes soos opgeloste suurstof, troebelheid, temperatuur, pH, spesifieke geleiding, chlorofille, nitraat en soutgehalte. Verskillende wiskundige en statistiese modelle is al gebruik vir die voorspelling van tydreeksdata. Masjienleer kan ook gebruik word as daar genoeg data beskikbaar is. In die besonder het kunsmatige neurale netwerke sukses behaal met die oplos van sulke probleme. Sulke netwerke is konseptueel eenvoudig en maklik om te implementeer. In hierdie tesis word 'n oorsig van twee neurale netwerkstrukture aangebied vir die voorspelling van waterkwaliteitsveranderlikes. In die besonder word meerlaag-perseptrone (MLP's) en lang-kortte-rmyn-geheue (*long short-term memory*, LSTM) netwerke aangebied. Eksperimente is uitgevoer op Hog Eiland waterkwaliteitsveranderlikes en die resultate van die modelle word met behulp van verskillende akkuraatheidsmetrieke vergelyk, soos wortelgemiddelde kwadraatfout. Daar word gevind dat LSTM beter presteer as MLP volgens meeste van die akkuraatheidsmetrieke.

Contents

Chapter 1

Introduction

Water is vital to all life. Human activities and chemicals may affect the physical, biological and chemical properties of water. These properties can be used to define water quality. This study applies machine learning methods to model and predict several important water quality parameter values (variables) over time. This chapter discusses water quality and its importance, as well as a number of the water quality variables and their relationships. Also, the research problem and objectives are presented.

1.1 Background

Water has three main sources: rainwater, surface water and groundwater [6]. Water pollution has numerous effects, including adverse consequences for the aquatic ecosystem. Polluted water resources may pose health risks. Therefore, it is necessary to ensure that the water quality meets specific standards. Water quality is often defined according to whether it is used for domestic, agricultural or industrial purposes. Depending on the purpose, different standards are applied. These standards are linked to the minimum chemical, physical and biological measures that water has to meet.

Many water boards monitor water variables to ensure that the quality of the water complies with specified standards. Institutes such as the World Health Organisation (WHO) specify standards that are used to determine the designated use of the water [53].

The importance of water quality variables encourages several institutes to record their values. Institutes like the United States Geological Survey, the San Francisco Department of Public Health, and the State Water Resources Agency of

Ukraine offer up-to-date water quality datasets[1], because it may be useful to analyse and learn from the historical data. A possible application could be to use the data for the prediction of future water quality variable values.

1.2 Water Quality Variables

The assessment of water quality demands measuring different parameters or variables which should be compared to the minimum standards for its designated use. This section presents some of the variables used in defining water quality.

1.2.1 Dissolved Oxygen

Dissolved oxygen (DO) is the amount of gaseous oxygen (O_2) dissolved in the water. Oxygen is absorbed from the atmosphere by water movement over rocks and waterfalls. DO in fast-flowing water is higher compared to standing or slow-moving water. Another source of oxygen in the water is the photosynthesis process of algae and aquatic plants. This increases the level of DO during the day, while at night or on cloudy days, the level of DO drops due to consumption of the oxygen by the algae and aquatic plants. DO is one of the most critical factors affecting aquatic organisms. Its concentration from the water surface to the bed decreases and limits the types of organisms. DO is reported in milligrams per litre (mg/L) or parts per million (ppm). Water containing 5 to 10 ppm is generally of acceptable quality and can enable a balanced aquatic ecosystem [10].

1.2.2 Turbidity

Turbidity is related to the cloudiness of water caused by particles that are, generally, invisible to the eye. Polluted water will have high turbidity, which may be due to phytoplankton or human activities. The existence of such contaminants in the water impacts the health of humans, animals and plants. Water pollution can be lessened by applying treatment to the water, or effluents before discharging to the water bodies [10].

Turbidity is measured in the Formazin Turbidity Unit (FTU), or the Formazin Nephelometric Unit (FNU). There are several methods to measure water turbidity, such as visual methods (using a Secchi disk, for example) or full-scale meters like turbidimeters [10].

[1] https://www.usgs.gov/, https://sfwater.org/index.aspx, https://www.davr.gov.ua/

Figure 1.1 illustrates the process of adsorption and desorption of chemicals in the water. This results in changing the turbidity of water. The decay process is proportional to turbidity.

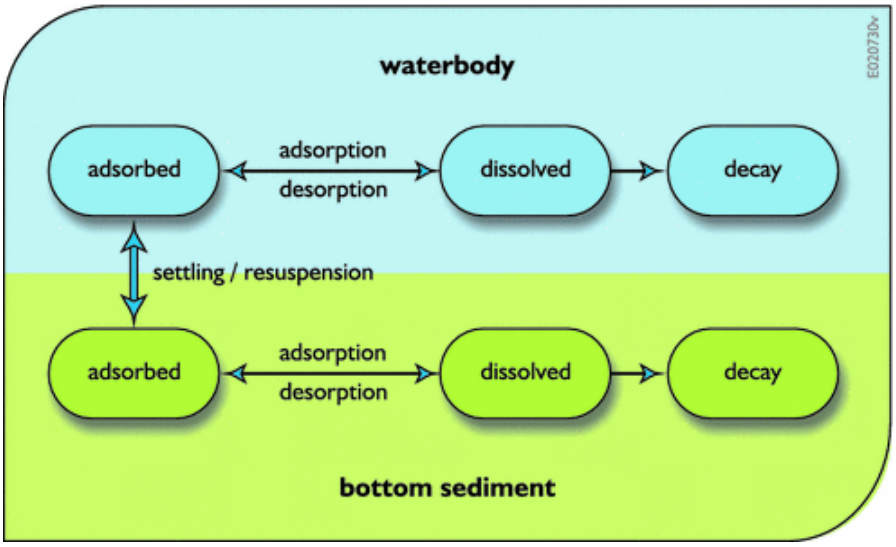


Figure 1.1: Schematic of the adsorption/desorption and decay processes of various toxic chemicals in water bodies and bottom sediments [43].

Turbidity is inversely proportional to DO. Decomposers such as bacteria use high amounts of oxygen to decay substances in the water.

1.2.3 Temperature

Temperature is the heat level of the water. Heat sources are shortwave solar and longwave atmospheric radiations. Water temperature decreases by evaporation, conduction as well as longwave radiation emission. The ability of gas substances like oxygen to dissolve in water decreases as temperature increases. Warm water may not have sufficient oxygen needed by the aquatic ecosystem, and water used in cooling electrical power for industry use, for example, has to be cooled to drop the temperature.

1.2.4 pH

pH stands for potential of hydrogen and is a chemical measure of the hydrogen ion concentration of water. The pH scale measures how acidic or basic a substance is. This scale ranges from 0 to 14. Values below 7 are considered acidic,

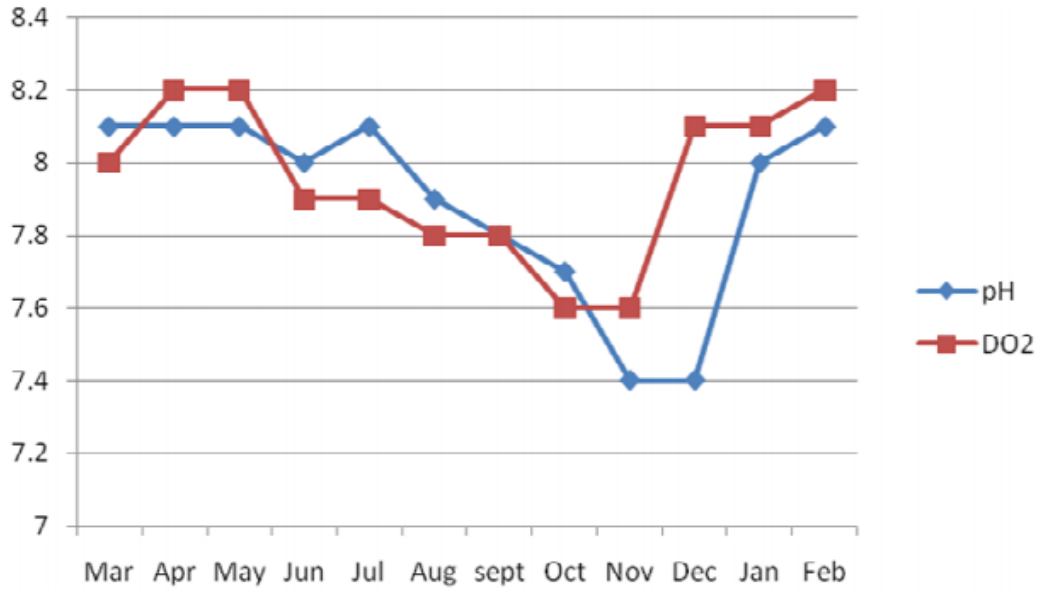


Figure 1.2: A comparison of pH and DO levels in Lake Asa in Nigeria [13].

values higher than 7 are basic and a value of 7 is neither acidic nor basic, but neutral. Pure water has a pH value of 7. These numbers are reported in logarithmic units and represent a 10-fold change in acidity or basicness of water. The pH determines which organisms can live in that water. pH is affected by several factors, such as carbon dioxide. When carbonic acid increases, the level of pH decreases [5].

pH in the water is highly correlated to DO, as illustrated in Figure 1.2. As the values of pH increase in Lake Asa from December to February, the values of DO also increase. The same effect happens when pH decreases from June to October.

1.2.5 Specific Conductance

Specific conductance (SC) is the ability of water to conduct an electric current for a unit length. It is an indirect measure of the presence of dissolved solids in the water. When nitrate, sulfate or phosphate compounds break down, more ions are found in the water. These ions, which are negatively or positively charged, increase water conductivity.

Conductivity is also affected by temperature: An increase in water temperature increases the conductivity. Therefore, SC is reported at 25 degrees Celsius and expressed as Siemens per centimetre. Pure water has 0 S/cm. Rainwater has a

higher SC value due to dissolved gases from the air, dust particles and airborne material. Seawater can reach approximately 50000 S/cm because it contains high amounts of dissolved salts. Since SC indicates the number of solids in water, it can be used to detect pollutants [10].

1.2.6 Chlorophylls

Chlorophylls are green pigments found in algae and plants that allow photosynthesis. Although algae and aquatic plants are essential in water, a too high concentration decreases the DO levels, which results in poor water quality. The increment of algae and aquatic plant biomass is measured by the concentration of chlorophyll, which indicates a degraded water quality.

Chlorophyll concentration is measured using a fluorometer, which reads the amount of transmitted light when water samples are exposed to a specific wavelength. Satellites are also used to measure chlorophyll by imaging and reading the changes in the colour of the water source [10].

1.2.7 Nitrate

The nitrate ion (NO_3) is a common form of nitrogen found in natural water. Nitrite (NO_2) and ammonia (NH_3) are other forms. Figure 1.3 shows the different forms of nitrogen.

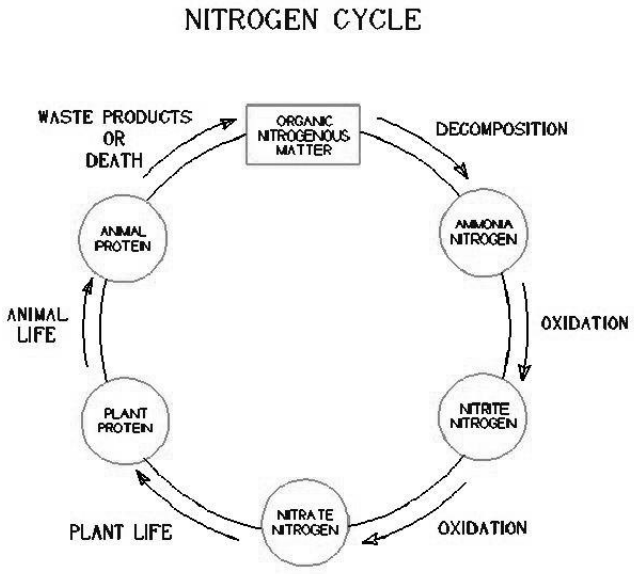


Figure 1.3: Nitrogen is continually recycled by plants and animals [3].

Nitrogen is an essential constituent of proteins and genetic material in living organisms. The NO_3 in water stimulates algae and plant growth. However, an excessive concentration of NO_3 in the water can cause plankton over-production. When they die, oxygen is used by the decomposition process, which can be hazardous to living organisms that depend on oxygen, and also may affect human health. Nitrogen enters the water through municipal and industrial wastewater, animal wastes and fertilisers that are carried by stormwater runoff. Stormwater runoff may result in an increased nitrate concentration in farming areas and their water sources [10].

1.2.8 Salinity

Salinity is the amount of salt in water, and differs based on the source. Oceans have a higher salt percentage than rivers and streams. The tidal cycles and water flow fluctuate the salinity levels in water periodically. Higher temperature increases water evaporation which results in a higher salinity. Salinity is inversely proportional to DO. The salinity distribution can also affect organisms. Some animals can be affected negatively in terms of growth reduction and reproduction limitations [10].

1.3 Problem Statement

Many factors influence water quality. Measuring water quality variables has become increasingly important to determine the quality of the water, for comparison to certain standards (see Table 1.1). The measuring process can be labour intensive, costly and time-consuming. However, several institutes have been recording measurements for long periods at different stations. Water authorities use several models to predict water quality, but these models are relatively simple compared to the complexities of actual water systems. Therefore, it is necessary to have models that overcome these limitations.

Machine learning (ML) can potentially be used to predict the future values of variables from recorded historical data. Several basic ML methods have already been used to predict and forecast quantitative characteristics of water. Such predictions are aimed at providing water researchers with a better grasp of how various characteristics relate to each other. However, available data is usually in the form of time-series, and it seems promising to investigate ML models that account for time dependencies. Therefore multilayer perceptrons (MLPs) and long short-term memory (LSTM) networks are useful to be applied. Moreover, ML models are highly dependent on data availability. For measuring stations where data is abundant, ML models can learn effectively from historical data,

Table 1.1: Water quality variables and their standards[1] for domestic, industrial and agricultural use.

Variable	Unit	Domestic	Industrial	Agricultural
pH	-	6.0 - 9.0	7.0 - 8.0	6.5 - 8.4
Nitrogen	mg/L	-	-	5
COD [2]	mg/L	-	0 - 10	-
Chlorophyll	mg/L	0 - 1	-	-
Chloride	mg/L	0 - 100	0 - 20	100
Salinity	PSU	47.92	6708.33	0.000003
Turbidity(Color)	Pt-Co	15	-	-
Nitrate	mg/L	0 - 6	-	0.5

[1] The standards in the table are defined by the South African Water Quality Guidelines. The presented values are taken from [23], [24], [22].
[2] COD stands for chemical oxygen demand.

and where data is sparse, transfer learning can be applied. Transfer learning is the ability to leverage knowledge gained from one ML model in another.

To summarise, the main problem that this thesis considers is to find optimal machine learning models that give high accuracy in modelling and predicting time-series data of water quality variables.

1.4 Aims and Objectives

This study aims to achieve the following:

I. Analyse water quality variables. The analysis process involves:
   i. An investigation into cross-correlation between different variables. This investigation helps in identifying the relationship between variables and gives an insight of which variables can be used to predict others.
   ii. An investigation into auto-correlation. This investigation shows how many time steps should be taken into account when predicting the variables.
   iii. Identify missing values and anomalies in the data and apply replacement methods.

II. Predict future values of the variables using machine learning. Multilayer perceptron and long short-term memory models will be applied.

III. Compare results of the prediction models to determine optimal model parameters.

1.5 Thesis Outline

The rest of this thesis is organised as follows. Chapter 2 reviews previous works that apply machine learning techniques for water quality prediction. Chapter 3 presents a detailed description of the methods applied in the study, which include multilayer perceptrons and long short-term memory networks. The chapter also provides reasons for choosing these models. Chapter 4 contains the research methodology. Chapter 5 shows the results of the experiments and model evaluations. Finally, Chapter 6 concludes and presents potential future work.

Chapter 2

Literature Review

This chapter explores works that have been done in water quality (WQ) prediction. Most of the studies found in this domain focus on either predicting the values of water quality variables or classifying the water quality into different levels based on the values of the water quality variables. Limited water quality data and the high cost of water quality monitoring often pose severe problems for process-based modelling approaches.

A subset of literature has been selected based on the model that the study uses. The review is divided into three main categories: mathematical, statistical and machine learning models. The main difference between them is the method they used. Mathematical models use physics-based equations to determine water quality or to forecast water quality variables. Statistical models use statistical measurements like the mean, variance, and distribution of the data when defining the quality of water. Finally, machine learning models learn from historical data to make predictions or to quantify the level of water quality. More details will be discussed in the following sections.

The selected studies cover region and the time of the study, the water quality variables considered, the methods used in defining water quality, the architecture of the model under study, the accuracy of the developed models and the study limitations. Some of the technical terms mentioned in this chapter will be covered in more detail in Chapter 3.

Table 2.1 presents a summary of the covered studies. Sections 2.1, 2.2 and 2.3 present more details about them.

Table 2.1: Comparison of different machine learning techniques for evaluating WQ and its variables.

Author	Model	Location	Goal	Variables	Data points	Architecture	Accuracy
Palani [54]	GRNNs	Singapore coastal water - East Johor Strait	To predict weekly values of WQ parameter from historical data	salinity, temperature, DO, and Chl-a	48	3 hidden layers Activation function: G, tanh and GC Initial weight: 0.3 learning rate: 0.1 Momentum: 0.1	Temperature:MSE<0.5, R^2>0.7, Salinity:MSE<1.3, R^2>0.66, DO:R^2=0.95,MSE =0.28, Chl-a: RMSE 23.6 to 1.33 and R^2 0.99 to 0.51
Chine [19]	BPNNs	Northern Taiwan - Feitsui Reservoir	Predict TP monthly values	TP	96	Input: 1 Hidden: 4 Output: 1 Iterations: 2000	RMSE training: 9.43 RMSE testing: 13.12
Zhu and Hao [70]	FNN and PNN	Suzhou, China	Rank the quality of the water in 5 grades	CODcr, BOD5, DO, and NH3-N	Not given	Not given	Not given
Lu and Huang [44]	Decision Tree	North China	Predict Chl level of coming day in the water	Chl, DO and solar radiation	10810	Not given	90% (metric is not given)
Muhammad [46]	ANN, Naive Bayes, Kstar and J48	Kinta river, Perak, Malaysia	Identify the most significant features contributing to water classification	DO, COD, BOD and 50 other parameters	7155	Not given	Naive Bayes: 85.19% (metric is not given)
Khan and See [35]	NAR	New York, US	Predict the values of WQ parameters the next day	Chl, SC, DO and Turbidity	52560	Not given	Testing 0.78-0.99, 0.00033-0.0022 and 0.0181-0.047 for R^2, MSE and RMSE respe--ctively.

Comparing the studies and their results is a good way to identify potential gaps. The gaps in some of these studies have motivated the objectives of this thesis.

2.1 Mathematical Models

Xing et al. [67] and Banerjee and Srivastava [14] applied mathematical models to evaluate the quality of the water in different areas in China. Although the two studies were carried out in the same region, the models were applied to different water sources. Xing et al. [67] used fuzzy comprehensive evaluation (FCE) based on entropy weight (EW) to identify the quality of underground water. They studied calcium carbonate, total dissolved solids, chloride and sulfate variables. FCE-EW starts by defining the WQ variables of interest and evaluation grade set. The grade set is defined based on ground WQ standards. The last step of this model is to apply the entropy weight method. On the other hand, Banerjee and Srivastava [14] used a comprehensive water quality identification index (CWQII) to classify water into five classes based on its quality. CWQII is an algebraic formula to analyse the variables quantitatively to determine the difference in values. CWQII was calculated based on the single factor water quality identification index (SFWQII) values, and the study evaluated dissolved oxygen, ammonia, chemical oxygen demand, and total phosphorus.

Banerjee and Srivastava [14] used three years of data, unlike Xing et al. [67] who used only a small number of observations. The accuracy of the models was, however, not discussed.

The dissolved oxygen variable used by Banerjee and Srivastava [14] will also be investigated in this study.

2.2 Statistical Models

Two statistical methods by Li and Wang [41] and Photphanloet et al. [55] are reviewed here.

Li and Wang [41] used a Bayesian model to assess the quality of water in three locations in the Gorges reservoir. They examined oxygen indexes (DO and COD), nutrient salt (TP and TN) and poison indexes (Cu). Although it was a simple and effective method with low computational complexity, the study did not report the accuracy of the model in the assessment process.

Photphanloet et al. [55] applied alpha-trimmed auto-regressive integrated moving average (ARIMA) to predict the upcoming biochemical oxygen demand (BOD) data using past data in four monitoring stations along the Chaophraya

River of Thailand. The model was applied on both seasonal and non-seasonal time-series data collected between 1996 and 2013 at 18 stations by the Pollution Control Department, Ministry of Natural Resources and Environment. The study focused on only four stations. The trimmed ARIMA error percentage was compared to the smoothing rate, and the results of applying these models were 6.30%, 18.97%, 6.12% and 28.44% compared to the 38.75%, 39.46%, 13.67% and 44.74% errors of the smoothing method.

As Li and Wang [41] examined the DO and nitrate, this study will also cover both.

2.3 Machine Learning Models

Several machine learning models have been applied in determining the quality of water. They mostly make use of some forms of neural network. This section covers some of the studies in this area.

Five out of six machine learning methods found to predict water quality used artificial neural networks (ANNs) or one of its variants. Several studies have applied regular backpropagation neural networks with different architectures ([54], [19] and [46]). Zhu and Hao [70] and Khan and See [35] used fuzzy neural networks (FNNs) and nonlinear auto-regression neural networks (NARNNs), respectively. These models are manipulated versions of ANNs.

The study by Palani [54] used ANNs to predict the weekly values of salinity, temperature, DO, and chlorophyll-a in East Johor Strait coastal water, Singapore. The data was collected between December 1996 and June 1997 [20]. The collected data were for four stations numbered as 1, 2, 2alt and 3. Station 1 and 3 have 32 data points in total, and they were split into 80% training 20% testing. Station 2 and 2alt have 16 data points, which were used as a validation set. The ANNs have three hidden layers which have Gaussian, hyperbolic-tangent, and Gaussian-complement as activation functions.

The researchers noticed that fewer hidden nodes are better for the model because it is useful for generalisation and is less prone to over-fitting. Weights of the nodes are generally distributed from -1 to 1. The initial weight is 0.3 for the nodes, and the optimal learning rate is 0.1 and momentum is 0.1. The model was trained to predict two time steps into the future. Using insights of the data correlation, the model takes temperature, salinity, and DO values to predict temperature, salinity, and DO respectively.

Applying this model on up to two time steps in the future results in different accuracies measured using mean squared error (MSE), root mean squared er-

ror (RMSE), mean absolute error (MAE), and R^2 (regression score). The results were: MSE < 0.5 and R^2 > 0.7 for temperature prediction, MSE < 1.3 and R^2 > 0.66 for salinity prediction, and R^2 = 0.95 and MSE = 0.28 for DO prediction. Lastly, chlorophyll-a was predicted with different architectures and other variables as inputs, and the accuracy varied from 0.99 to 0.51 for R^2 and 23.6 to 1.33 for RMSE. This study was limited by the dataset size and could provide more valuable predictions if the data size was big enough.

Chine [19] used ANNs to predict the concentration of the total phosphorous (TP) at Feitsui Reservoir in Northern Taiwan. Monthly monitoring data records from 1996 to 2003 were obtained from the Feitsui Reservoir Administration Bureau and used in the study. The first six years of data were used in training the model while the last two years was used in testing. The ANN model consisted of three layers that have one node, four nodes and one node in the input, hidden, and output layers. The network was trained for 2000 iterations.

The RMSE was calculated to measure the accuracy over the training and testing set. The RMSE reached 9.43 for the training set and 13.12 for the testing set. The study did not make use of any other parameters to define the phosphorous concentration and did not consider a validation set.

Although Zhu and Hao [70] and Lu and Huang [44] both aimed to predict the quality of the water in China, and their studies were carried out in the same year, Zhu and Hao [70] used FNNs while Lu and Huang [44] used decision trees. The aims of these two studies were also different. The first one aimed to rank the quality of water, while the second performed a prediction model to find the level of chlorophyll in the coming day.

Zhu and Hao [70] used data between 1999 and 2002. In this study, chemical oxygen demand (COD), 5-day biochemical oxygen demand (BOD5), DO and nitrogen-ammonia (NH_3-N) variables were used as evaluation indicators. The quality of the water was divided into five grades according to the Surface Water Environmental Quality Standard (GB3838-2002) issued by the government of China. Zhu and Hao [70] structured their FNNs to contain five layers: input, fuzzification, fuzzy reasoning, reconciliation and output. The FNNs were able to assess the concentrations that defined the water quality. Mean squared error (MSE) was used to measure the accuracy of the model, but the researchers did not publish the results. Zhu and Hao [70] claimed the reliability and effectiveness of the FNNs models not only in predicting river WQ but also ground WQ and the prediction of quality in the atmospheric environment. The latter was not motivated clearly.

Lu and Huang [44] used chlorophyll (Chl) to quantify the existence of algae

in the water. The forecasting process used Chl of a previous time, dissolved oxygen (DO) and solar radiation. Lu and Huang [44] used a dataset of 116 days, with 94 values per day. The decision tree model has to divide these values into four Chl levels, by considering the Chl values, the standard deviation of DO and the mean value of the radiation. The model uses only 100 days for training and 15 for testing (1 day's data was missing). The model predicts 87 data points effectively, which results in 90% accuracy. The accuracy percentage was calculated based on comparing results of predicted and actual values of the Chl. Despite the high accuracy of the model, the training process does not cover all of the tree bifurcations. The dataset used by Lu and Huang [44] was small and did not represent all seasons.

More recent studies found on water quality prediction were performed by Muhammad [46] and Khan and See [35]. Both studies applied ANNs in different areas. Muhammad [46] applied not only ANNs but also naive Bayes, Kstar and J48 methods to classify surface water quality in Malaysia. Khan and See [35] applied nonlinear auto-regression neural networks to predict water quality in the United States. Muhammad [46] identified the best model as well as the significant variables that affect water quality. The dataset was obtained from the East Coast Environmental Research Institute in the University of Sultan Zainal Abidin, which contained monthly records of 135 instances and 54 attributes from 2002 to 2006. The variables include dissolved oxygen (DO), chemical oxygen demand (COD), biochemical oxygen demand (BOD), pH, and many others. Muhammad [46] selected 53 variables and applied the models to them. The accuracy results were as follows. The naive Bayes algorithm achieved the highest accuracy of 85.19%. However, Kstar had the highest accuracy (86.67%) when applied to the six variables DO, BOD, COD, SS, pH, and NH_3-N. Only 67.41% was the accuracy of the bagging method after applying it to classify NO_3, Ca, and water quality index class variables.

Muhammad [46] did not discuss the classification method or how the classification was done. Furthermore, the study did not explain the reason for choosing some variables over others in the classification process. In contrast, Khan and See [35] studied four variables: chlorophyll, specific conductance, DO and turbidity. The data was obtained from the United States Geological Survey (USGS) in 2014, with 6-minute time intervals, and further divided into 60% training, 20% validation and 20% testing data. The main goal of the study by Khan and See [35] was to predict future values of the variables based on their present values. The developed model consists of three layers: input, hidden and output, and was evaluated using MSE, RMSE and regression analysis (R^2).

The present study identifies the benefits of each previous study. Based on the

gaps in the literature, this study applies machine learning models, specifically ANNs (multilayer perceptrons and long short-term memory networks). Khan and See [35] inspired the present study and use the same source of data but over different periods.

2.4 Summary

Several methods have been used to evaluate and predict the quality of water. This chapter presented some of the mathematical, statistical and machine learning models that have been used in predicting water quality and its variables. Among these, statistical models are relatively simple and easy to implement. Most of the reviewed studies that applied machine learning models used neural networks and applied it on DO or oxygen-related variables.

ANNs provide a particularly good option since they are computationally fast and require substantially fewer input parameters and input conditions. ANNs, however, require a large pool of representative data for training. None of the reviews found in the literature applied long short-term memory models. The studies have considered data from different regions of the world, but none have considered Africa, which is likely due to data scarcity.

Chapter 3

Machine Learning Models

This chapter discusses the concept of machine learning (ML) and types of learning. ML has proved to be a good solution for predicting water quality variables using historical data, as discussed in the previous chapter. This chapter presents some of the ML models that have been used in the prediction of water quality variables, including artificial neural networks (ANNs) and also some of the models that are used in predicting time-series data, namely recurrent neural networks and long short-term memory models.

The chapter starts with an introduction to neural networks and their structure, as well as a description of the relationship between ANNs and biological networks.

3.1 Fundamentals of Machine Learning

ML is the field of study that gives computers the ability to learn without being explicitly programmed [50]. The machine adopts new mechanisms that were learned from experience, example, or analogy [49]. ML is a field of artificial intelligence that focuses on building smart systems such as robotics, pattern recognition systems, natural language processing systems and computer vision.

There are several different methodologies used in the learning process. Supervised and unsupervised learning are discussed in the following points.

- Supervised learning: Algorithms of this type train the system with the given input data as well as with the correct output. The system later associates its experience in predicting the output of new input data. This approach can solve regression problems where continuous values are predicted, as well as classification problems where categorical values are pre-

dicted [50]. An example of a regression problem is predicting the values of water quality variables the following day by using the current day's values.

- Unsupervised learning: This type of algorithm identifies structure out of the dataset by partitioning or clustering the data. Unsupervised learning can, for example, be used in classifying water quality variable values into normal and anomalous [50].

Each of these learning types has specific models used for solving its problems. One of the most famous ML models is the ANN. ANN models counterpart the biological brain and its functions [28], and it requires few input parameters and conditions; hence is computationally inexpensive. ANNs are mostly used in solving prediction and classification problems.

3.1.1 Neural Networks

A neural network is an interconnected network of relatively simple processing elements called neurons. A neuron is a cell in the brain that processes information [49]. This information is processed by a signal transformation from one neuron to another, and these signals are transmitted via synapses. Figure 3.1 is a schematic drawing of a neural network.

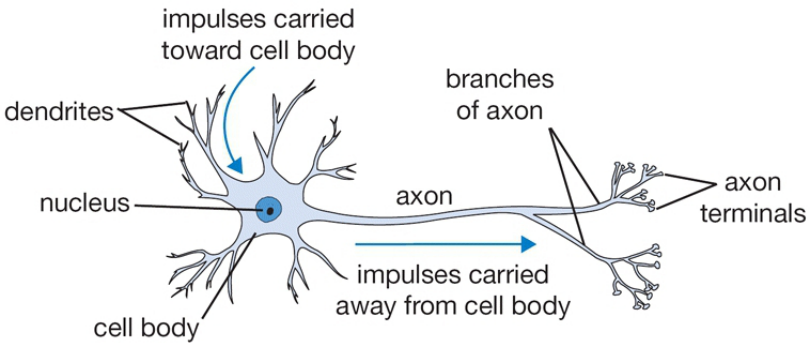


Figure 3.1: The biological neural network [34].

Electro-chemical reactions generate a signal in one neuron which is propagated to another neuron in the brain through synapses. If the signal reaches a threshold, it will be determined if a specific neuron has to transfer the signal or not [38].

The brain can be considered a highly complex, nonlinear, and parallel information-processing system, in the sense that data storage and its processing are done simultaneously throughout the network.

The idea behind ANNs is to model features of the brain and its ability to learn [15]. ANNs resemble the human brain in that this model can observe patterns in a sufficient number of samples and use this experience to generalise to other examples that have not yet been encountered. Despite the link between ANNs and the brain, ANNs do not simulate what the brain does.

Section 3.1.2 discusses a simplified version of the biological neuron, called a perceptron, that retains the general learning behaviour.

3.1.2 The Perceptron

A perceptron is a single artificial neuron. It is a simplified structure of a biological neuron. Figure 3.2 illustrates the mathematical model of a perceptron by the biological neuron analogy.

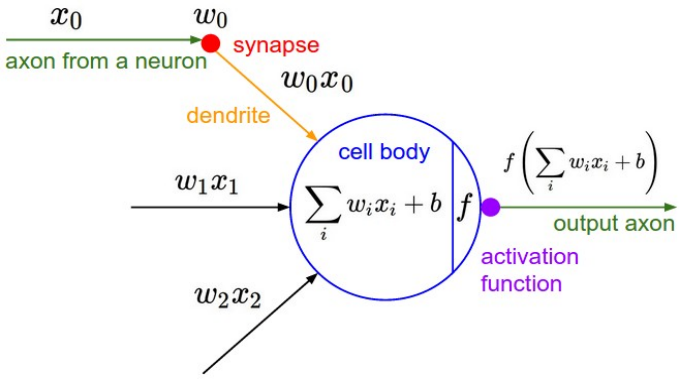


Figure 3.2: The structure of the perceptron [34].

A perceptron receives its input (x_i in Figure 3.2) from the dendrites and produces output along its axon (z). The basic function of a neuron is to add up the products of inputs (x_i) and their associated weights (w_i), then compare the weighted sum to a certain threshold. The perceptron will give an output only if the threshold is exceeded. The threshold is defined by the activation function (f), and more details of this function are given in Section 3.1.3. The weight expresses input efficiency to its corresponding output. The larger the weight, the more efficient the synapse will be and therefore more of the signal will be transmitted [15].

As seen in Figure 3.2, a perceptron firstly performs a linear mapping of the input values to produce a linear result from all input values, which can be any real numeric value. The final output is given by applying an activation function that converts the linear output to nonlinear output. The output of the activation function is either continuous or discrete (for example, a binary value) [52].

The operations associated with a perceptron are:

$$y = \sum_{i=1}^{N} w_i x_i + b, \tag{3.1}$$

$$z = f(y). \tag{3.2}$$

As observed in the steps of Algorithm 1, a perceptron applies to Euclidean instance space where $X \subseteq \mathbb{R}^d$. The algorithm updates the weights according to the difference between predicted and real values, where Δw_i is the weight update. This algorithm was modified from Agarwal [9].

Algorithm 1: Perceptron learning algorithm

input : d-dimensional training set $X = \{x_1, x_2, \ldots, x_N\}$
parameter : Initial weight vector $\mathbf{w} \in \mathbb{R}^d$

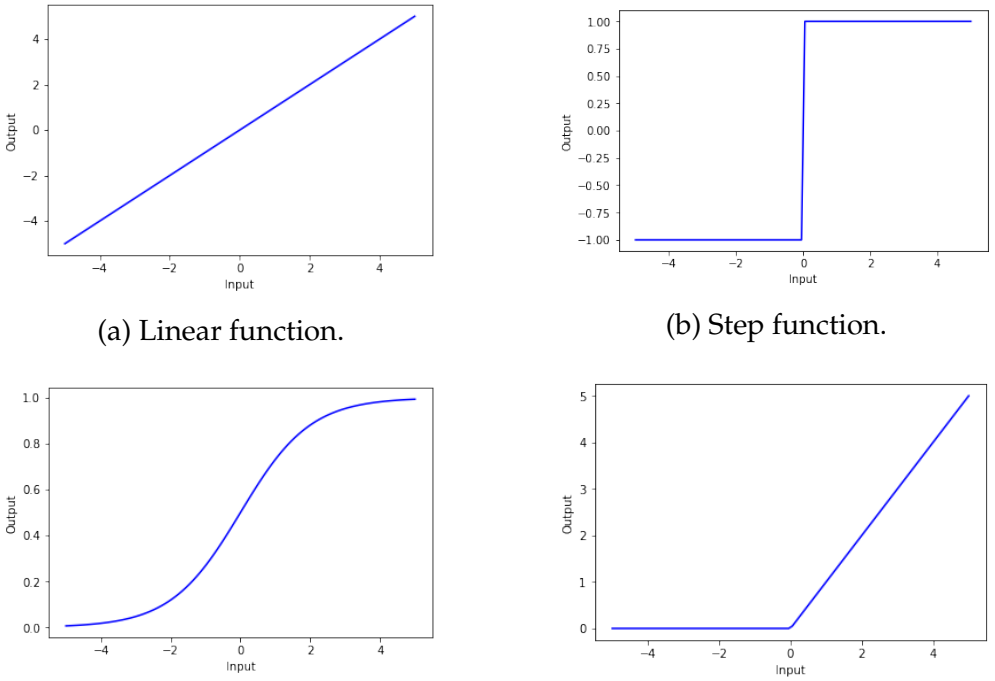
for *inputs and weights:* $i \in \{1, \ldots, N\}$ **do**
    $\hat{y} = f(\sum w_i x_i + b)$;
    receive the true value of y_i ;
    if $y_i \neq \hat{y}_i$ **then**
        $w_{i+1} \leftarrow w_i - \Delta w_i$;

3.1.3 Activation Functions

Several possible activation functions can be used in artificial neural networks. Activation functions decide if a perceptron will transform its value to the next level or not by acting on the weighted sum of the inputs and the biases [15]. Activation functions apply certain computations on the input data to produce an output. There are linear and nonlinear activation functions; the type of the function used depend on the problem. Selecting a suitable activation function can improve the result of the system [52].

Only a few of the activation functions are practical in certain situations. In one network, different activation functions can be used. An activation function may

perform differently based on its position in the network [52]. Some of the common activation functions presented in Figure 3.3 are discussed, starting with linear and followed by the most common nonlinear activation functions.



(a) Linear function.

(b) Step function.

(c) Sigmoid function.

(d) ReLU function.

Figure 3.3: Illustration of four of the activation functions used in ANNs.

- Linear: The linear function (also called identity function) is illustrated in Figure 3.3a. This function performs a linear transformation from input to output, where the output of the function is simply equal to its input:

$$z = y. \tag{3.3}$$

Nonlinear activation functions help in learning higher-order polynomials beyond first order, as well as complicated models. The output of the activation function is either passed to the output (in case of having only one layer of perceptrons) or to the next layer (if the network is multilayered). The nonlinear activation functions are differentiable (except the step function), so that they may be used in the backpropagation process [27].

- Step: The step function in Figure 3.3b is a hard limit function. The input of the function is compared to a threshold zero and the output is either one or negative one:

$$z = \begin{cases} 1, & \text{if } y \geq 0, \\ -1, & \text{otherwise.} \end{cases} \tag{3.4}$$

- Sigmoid, also known as the logistic or squashing function: Sigmoid is a nonlinear function commonly used in neural networks. Sigmoid takes real values and applies the following function:

$$z = \frac{1}{1 + e^{-y}}. \tag{3.5}$$

  The sigmoid output will be bounded between zero and one, as seen in Figure 3.3c. The output is used in probability prediction and has been applied in different areas like classification and logistic regression modelling [52]. A study by Glorot and Bengio [26] presents a few drawbacks of the sigmoid function, such as slow convergence. New activation functions were proposed to overcome those issues.

- Rectified Linear Unit (ReLU): This function has proven to be fast and successful in generalisation, which makes it widely used in many deep learning applications [58]. The representation of ReLU is as follows:

$$z = \max(0, y), \tag{3.6}$$

  and shows that it performs a rectification process by applying a threshold operation to its inputs. Input values less than zero are set to zero. This range reflects the linearity property in the ReLU function, which makes it easy to optimise over with gradient-descent methods [27].

  The ReLU function enhances the computation speed [69] and ensures sparsity of its output. However, ReLU can overfit compared to other functions like sigmoid. ReLU causes demise of some gradient effects since it squashes the input between zero and the maximum [27]. Applying regularisation techniques increases the efficiency of ReLU.

3.1.4 Artificial Neural Networks

ANNs consist of simple and highly interconnected perceptrons, which are analogous to the biological brain's neurons. Perceptrons are connected by weighted links to pass information, with each perceptron receiving several inputs and

producing only one output. This output branches and transmits to other perceptrons. An ANN engineers the biological neural network's function where somas are represented by perceptrons, dendrites by perceptron inputs, axons by perceptron outputs, and synapses by the weights [49]. An ANN consists of several layers: an input layer $X = \{x_1, x_2, \ldots, x_n\}$, one or more hidden layer of the form $H = \{h_1, h_2, \ldots, h_m\}$ and an output layer $Y = \{y_1, y_2, \ldots, y_l\}$. Each layer has one or more neurons. Neurons in different layers are connected with certain weights. The input and weights of a neuron contribute with a nonlinear function in the hidden layer to predict the output of a neuron. Each layer passes its output to the next layer until the information reaches the output layer [49]. ANNs learn by adjusting the numerical weights in the links that connect the neurons. Weights are initially assigned randomly [28].

An ANN architecture needs to be defined, which includes setting the number of layers and neurons per layer. This step is followed by weight initialisation and updating based on the training examples. An example of an ANN with one hidden layer is shown in Figure 3.4.

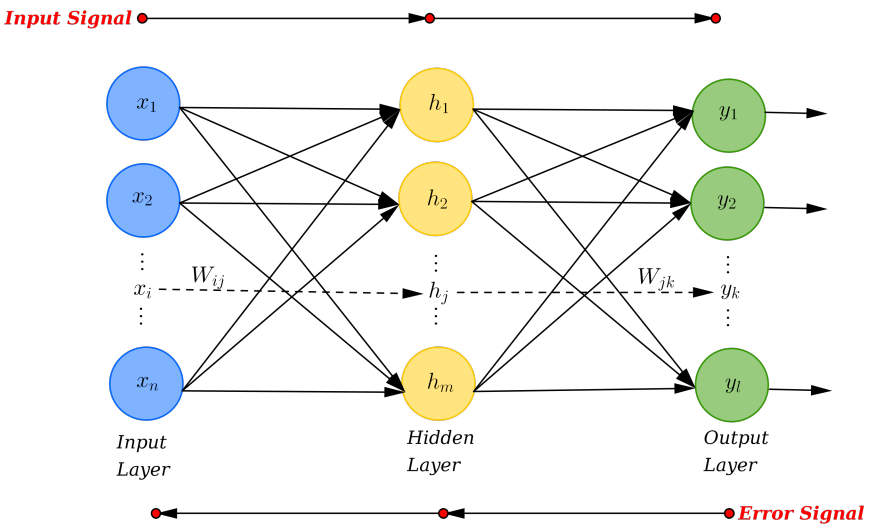


Figure 3.4: Input, hidden and output layers in an ANN.

To represent the ANN in a mathematical or computational form, matrices can be used since they offer a compact data structure.

An ANN computes the output by applying a function to the summation of the previous layer values and the values of the weights. Then, the difference between the ANN's predicted output and the actual value is calculated. The

generated error value is used to tune the weights between neurons. The main benefit of the tuning process is to find optimal weights in order to predict new output for unfamiliar inputs (to generalise). A learning rate (α) defines the rate of the learning process.

ANNs can either have a dynamic architecture like Elman models or static ones like multilayer perceptrons (MLPs). The most commonly used static ANN is the multilayer perceptron [28]. Therefore, this study will discuss and experiment with the MLP.

3.1.5 The Learning Process (Backpropagation)

Backpropagation is a method that is used in multilayer neural networks to define the error term in the network. Backpropagation follows the same procedures as the perceptron whereby it uses the output values, defines the error (the difference between the expected and actual values) and updates the weights. The error of the output layer is defined as:

$$e = y_a - y_p, \tag{3.7}$$

where y_a is the actual output and y_p is the predicted output. Updating the weights in the output layer of a multilayer neural network is similar to the perceptron whereby:

$$\Delta w_j = (\alpha)(\delta_k)(w_j), \tag{3.8}$$

$$\bar{w_j} = w_j - \Delta w_j. \tag{3.9}$$

The term δ in this case refers to the error gradient at neuron k. The error gradient δ is calculated by finding the derivative of the activation function multiplied by the error at the neuron output:

$$\delta_k = \frac{d}{dx_k} Sig(x_k)(e_k). \tag{3.10}$$

Finding the error term in a perceptron is relatively simple due to the structure of the sigmoid function. However, in more complicated cases, for example when multiple hidden layers are used, the error cannot be propagated unless differentiable functions are used. Differentiation of the sigmoid function gives:

$$\frac{d}{dx} Sig(x) = Sig(x)(1 - Sig(x)). \tag{3.11}$$

Thus δ_k is obtained as:

$$\delta_k = \underbrace{(y_k)(1-y_k)}_{\text{sigmoid derivative}} \underbrace{(y_{p_k} - y_{a_k})}_{\text{error at neuron } k}. \tag{3.12}$$

Weight correction at the hidden layer is the same equation that will be used to derive and update the weight change:

$$\Delta w_{ij} = (\alpha)(\delta_j)(w_{ij}), \tag{3.13}$$

$$\bar{w_{ij}} = w_{ij} - \Delta w_{ij}, \tag{3.14}$$

where δ_j is the error gradient at neuron j in the hidden layer and x_i is given by:

$$x_j = \sum_{i=1}^{n} x_i w_{ij} - b_j. \tag{3.15}$$

The error gradient δ_j at the hidden layer is also given by the same formula for calculating the error at the output layer:

$$\delta_j = \frac{d}{dx_j} Sig(x_j)(e_j), \tag{3.16}$$

where the derivative of the sigmoid function is:

$$\frac{d}{dx_j} Sig(x_j) = (y_j)(1-y_j). \tag{3.17}$$

However, δ_j is given by a different formula since the error is an accumulated gradient error for the previous layer's neurons:

$$e_j = \sum_{k=1}^{l} \delta_k w_{jk}, \tag{3.18}$$

where l is the number of neurons in the output layer. Thus the obtained δ_j will be:

$$\delta_j = \underbrace{(y_j)(1-y_j)}_{\text{sigmoid derivative}} \underbrace{\sum_{k=1}^{l} \delta_k w_{jk}}_{\text{error at neuron } j}. \tag{3.19}$$

ANNs are generally used with fixed input to generate fixed output, therefore, they do not capture the sequence of time-series data. The ANN architecture does not support non-consecutive perceptron interaction, for example, a perceptron at layer five only gets input from the perceptrons at layer four and six in the forward and backpropagation passes respectively. The recurrent neural network (RNN) is an adjusted model of the ANN that can be used in solving this problem. More details are given in Section 3.2.

3.1.6 Overfitting and Underfitting

The learning process of a neural network and its performance are highly dependent on the amount of the available data. Machine learning models and their variants may produce highly accurate results if they are provided with a large dataset.

If a model is provided with insufficient data in the learning process, then the model will fail to predict the training and testing datasets correctly. This problem is called underfitting. If the model is closely fitted to the training dataset and fails to generalise and predict new values, then it is called overfitting [51]. Several regularisation techniques have been proposed to avoid these problems. Some of them are discussed in the following section.

3.1.7 Regularisation Techniques

Regularisation techniques are processes that may help the model to generalise better on new data not seen during training. Some of the regularisation techniques are: L1 and L2 regularisation [39], batch normalisation [31], dropout [63], data augmentation [65] and early stoppage. This study focuses on and applies dropout and early stoppage techniques to avoid overfitting.

Dropout refers to the process of dropping out some of neurons from an ANN. This means temporarily removing some neurons with their incoming and outgoing connections [63]. A common dropout percentage is 0.5 which means the network should drop 50% of the neurons at random.

Early stoppage is the process of stopping the learning process before a minimum error on the training set is reached. It compares the validation error generated in the training at each step with the validation error at previous steps. If the validation error starts to increase then the model might be overfitting and training should stop [56].

3.2 Multilayer Perceptron

The multilayer perceptron (MLP) is an ANN with one or more hidden layers [18]. Each layer is a product function of $x_i \in X$ (which are the state variables) and $w_i \in W$.

The output layer produces a weighted sum of inputs added to a bias. Each hidden layer has weights and a bias. Nonlinear activation functions break the linear relationship between consecutive layers. Hence, all of the hidden layers behave independently and the relationship between successive layers helps in

understanding patterns in the data. However, the independence between neurons in an MLP leads the network to be incapable of conserving information about sequential events. The semantics of a typical MLP are shown in Figure 3.4.

The MLP model aims to optimise a loss function between the target $y_i \in Y$ and the predicted values $\hat{y}_i$. A common lost function is called mean squared error which is given by:

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{3.20}$$

There are many learning algorithms that can be used in the training of an MLP, of which the most common is backpropagation (as outlined in Section 3.1.5). The learning process in an MLP has two paths: forward propagation which is responsible for generating the output by applying linear and nonlinear functions to weighted sums of the inputs, and backpropagation which updates the weights after calculating the performance error. The forward pass is repeated to generate new errors and calculate new weights from the backward pass. As a result, the error is expected to reduce with every iteration. This process is repeated iteratively until convergence or a maximum number of epochs is reached [42]. Algorithm 2 explains how learning in an MLP works in a forward and backward pass.

A major issue facing multilayer neural networks is defining the error term. Backpropagation can be used to identify the error and adjust the weights. Section 3.1.5 discussed the process of backpropagation.

ANNs can fail to properly model sequential events in a time-series. A variant of ANNs called recurrent neural networks (RNNs) can handle this type of data, and is presented in the next section.

3.3 Recurrent Neural Networks

In Section 3.1.4 it was mentioned that ANNs learn from experience and prior observations. The power of these models is in their learnable weights which are used in processing inputs. Section 3.2 mentioned the potential problem when attempting to model sequential data with an MLP. RNNs are a special type of ANNs that were developed to model sequential data. RNNs have recurrent hidden states (internal connections between perceptrons) that use the current observations and hidden state vectors of the previous steps to define their output [40]. An RNN connects its perceptrons along the sequence, and these con-

Algorithm 2: MLP learning algorithm

input : d-dimensional training set $X = \{x_1, x_2, \cdots, x_N\}$, where $x_i \in \mathbb{R}^d$

parameter : Maximum number of iterations N and error rate e

```
while Iteration count < N and error function > e do
    for (i ← 1; i ≤ n_layers; i ← i + 1) do
        for (j ← 1; j ≤ n_neurons; j ← j + 1) do
            calculate weight sum;
            add threshold;
            apply activation function;
    for ∀ node in output layer do
        calculate error;
    for ∀ node in ∀ hidden layer do
        calculate node error;
        update node weight;
    calculate global error (error function);
    increase iteration number by 1;
```

nections demonstrate the dynamic behaviour of temporal data, thus leveraging long-term dependencies [47].

Figure 3.5 illustrates an RNN block and its components. The block takes two inputs and outputs two values.

RNNs can be used in modelling sequential data problems, for example speech recognition (sequence-to-sequence), music generation (integer-to-sequence), sentiment classification (sequence-to-integer), DNA sequence analysis (sequence-to-sequence), machine translation (sequence-to-sequence) and video activity recognition (sequence-to-integer). Many machine learning problems use sequences of input and/or output.

Section 3.3.1 describes the components of an RNN. RNNs have some advantages and disadvantages, with some of the most prominent listed in Table 3.1. The advantages of RNNs cause them to be better models than regular ANNs for predicting time-series data. Possible improvements to avoid the disadvantages of RNNs may result in a higher prediction accuracy.

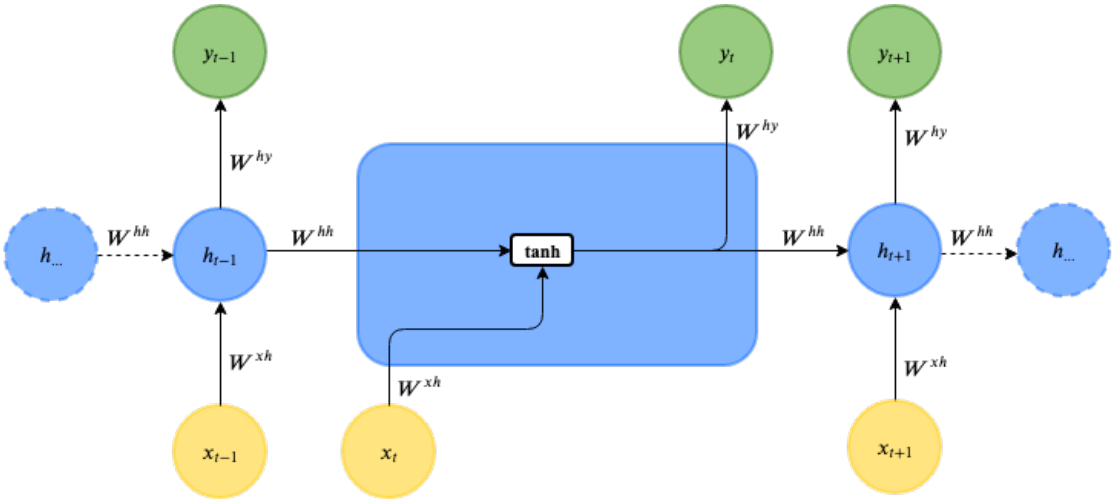


Figure 3.5: An RNN block.

Table 3.1: Advantages and disadvantages of RNNs, modified from Amidi and Amidi [11].

Advantages	**Disadvantages**
No input length limit	Slow computation rate
Weights are shared across time	Difficulty of accessing information
Data can be stored and used	Future inputs are not considered

3.3.1 Recurrent Neural Network Architecture

Like ANNs, an RNN consists of input, output and at least one hidden layer that can be connected in several forms [27]. The hidden layer has a self-loop called the recurrence. Figure 3.6 illustrates the folded and unfolded RNN structure. The unfolded form of the RNN shows the connections between perceptrons. The hidden states serve as a memory to keep information from previous inputs.

Every water quality variable series is given by vector $\{x_1, x_2, \ldots, x_N\}$ and $I,H,O \in \mathbb{N}$ are the sizes of input, hidden and output vectors, respectively. Thus, $x_t \in \mathbb{R}^I, h_t \in \mathbb{R}^H, \hat{y}_t \in \mathbb{R}^O$ and $y_t \in \mathbb{R}^O$ denote the input, hidden, output and target vectors at time t, respectively. Furthermore, let L_t represent the loss at step t which is the difference between the predicted value by the network $\hat{y}_t$ and the target value y_t. The overall loss is denoted by L. The weight matrices W^{xh}, W^{hh} and W^{hy} are shared across all time steps at each layer, where:

- $W^{xh} \in \mathbb{R}^{H \times I}$ connects the input and hidden state,

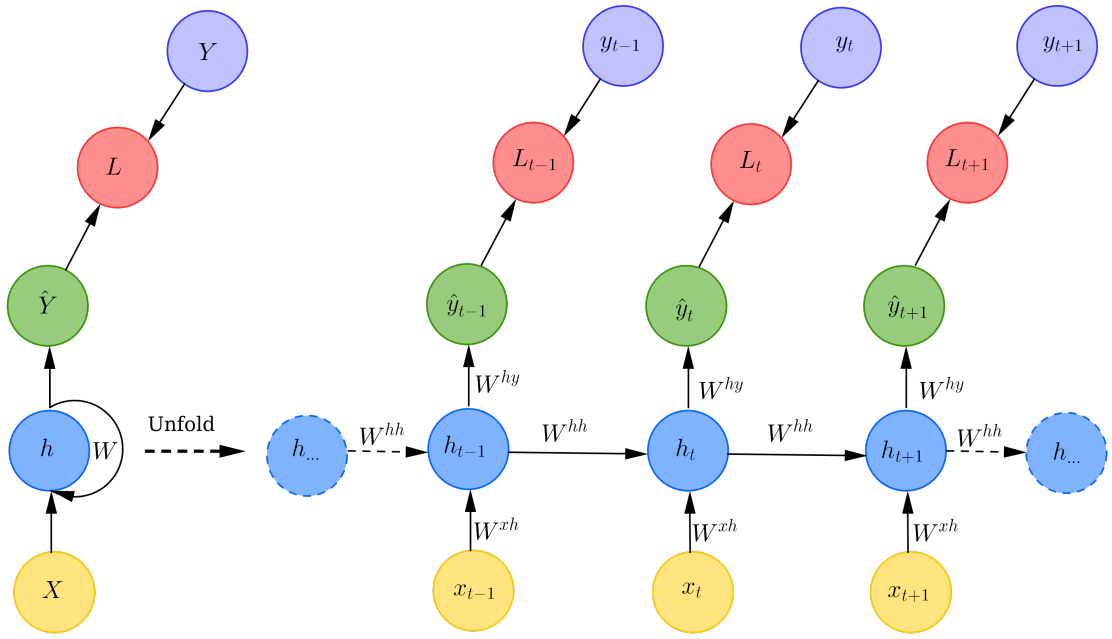


Figure 3.6: An unfolded single hidden layer RNN architecture with a loss.

- $W^{hh} \in \mathbb{R}^{H \times H}$ connects two consecutive hidden state,
- $W^{hy} \in \mathbb{R}^{O \times H}$ connects the hidden and output state.

The main goal of training an RNN is to find the optimal weight matrices that minimise the overall loss. Then the output of the network at time t will be computed as:

$$y_t = W^{xh} x_t + W^{hh} h_{t-1} + W^{hy} h_t + b, \tag{3.21}$$

where b is the bias vector. RNN also applies an activation function in each step after calculating the summation. During training, the weight matrices are updated using backpropagation through time, which will be discussed in the following section.

3.3.2 Backpropagation Through Time

The backpropagation through time (BPTT) method applies the same concepts as regular backpropagation, but with small adjustments [66]. This method applies the chain rule to find the gradients of the loss function. The loss at time t is given by:

$$L_t = \frac{1}{2}(\hat{y}_t - y_t)^2, \tag{3.22}$$

with the derivative:

$$\frac{\partial L_t}{\partial y_t} = y_t - \hat{y}_t, \tag{3.23}$$

where y_t is the expected observation and $\hat{y}_t$ the predicted observation. The total loss is given by:

$$L = \frac{1}{T}\sum_{t=1}^{T} L_t. \tag{3.24}$$

The gradient of the loss function with respect to W, where $W \in \{W^{xh}, W^{hh}, W^{hy}, b\}$ is given by:

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial W}, \tag{3.25}$$

where $\frac{\partial L_t}{\partial W}$ can be computed by the following chain rule:

$$\frac{\partial L_t}{\partial W} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial y_t}\frac{\partial y_t}{\partial h_t}\frac{\partial h_t}{\partial h_k}\frac{\partial h_k}{\partial W}. \tag{3.26}$$

In this case, $\frac{\partial h_t}{\partial h_k}$ represents the partial derivative of h_t with respect to all hidden layer neurons in previous steps, whereas k is given by $\{1, 2, \ldots, t-1\}$. The chain rule defines how the weight at step k affects the loss function at step t. The components for which $t \gg k$ and $t \ll k$ are called long and short term dependency, respectively. Figure 3.7 illustrates the BPTT process that is applied to the unfolded RNN model.

As the output calculations flow from the left to the right in the RNN architecture, the backpropagation flows in the opposite direction. In BPTT each perceptron gets two inputs: one propagates from the output of that perceptron and one from the previous perceptron.

3.3.3 Types of Recurrent Neural Networks

The input and output of RNNs govern the number of parameters RNNs need to perform its task. Some RNN types are listed below:

- one to one (Figure 3.8a)
- one to many (Figure 3.8b)

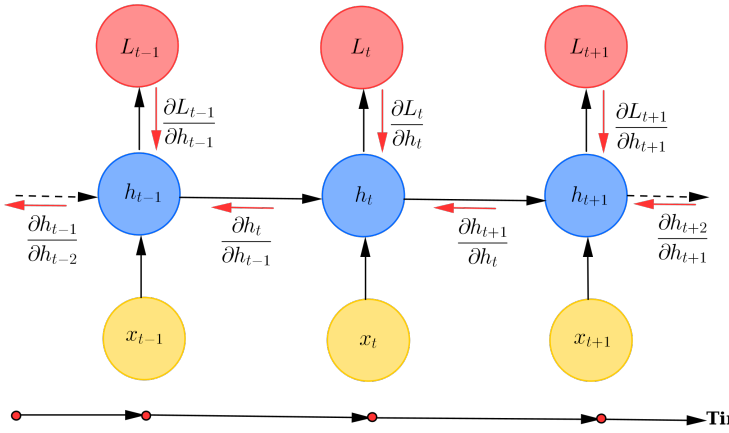


Figure 3.7: Backpropagation through time (BPTT) in an RNN model.

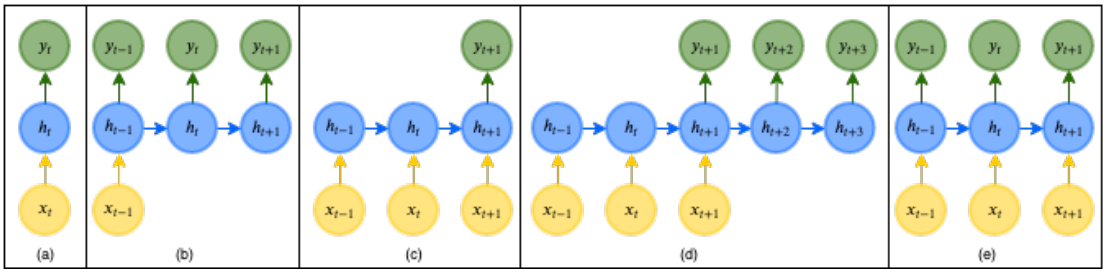


Figure 3.8: Different RNN structures based on the input and output.

- many to one (Figure 3.8c)
- many to many: output shifted (Figure 3.8d) or parallel (Figure 3.8e) to the input.

The structure of the RNN depends on its input and output. The many to many type has two structures, and both depend on the region to be predicted. This study focuses on one to one, and many to many types.

3.3.4 Exploding and Vanishing Gradients

Although RNNs have proved to be suitable for modelling time-series data with short-range sequences, it is difficult to train them for long-range sequences [16]. When backpropagation is applied through a long RNN, the error will either explode or vanish [29]. This problem is solved by introducing a cell state to the RNN, with the concept of gates. Section 3.4 covers this new model.

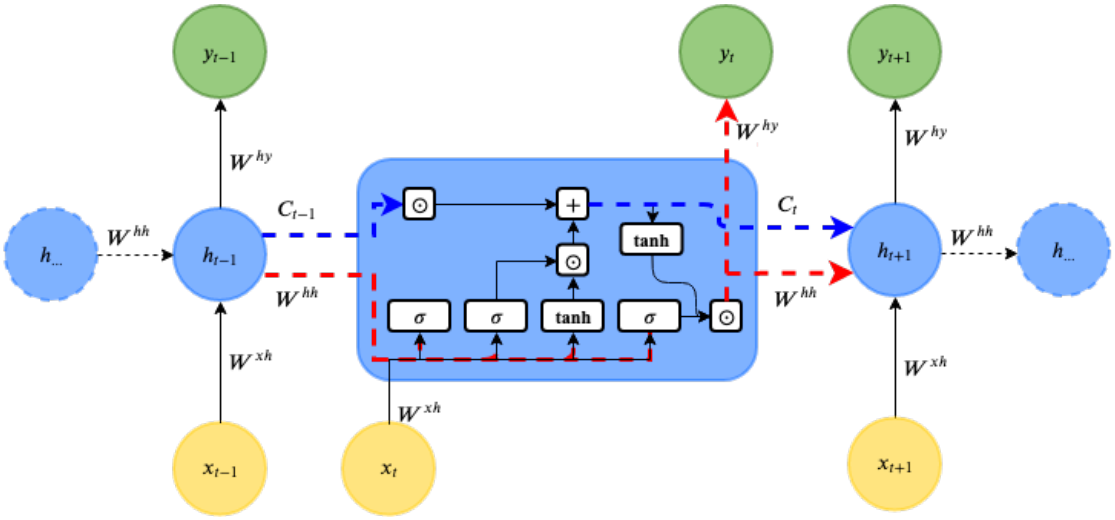


Figure 3.9: LSTM memory cell structure.

3.4 Long Short-Term Memory Networks

RNNs can in principle use their feedback connections to store representations of recent input events in the form of hidden states. Long short-term memory (LSTM) networks attempt to solve both exploding and vanishing gradient problems of RNN models. LSTM replaces the hidden vector of an RNN with memory cells, with gates to selectively remember information in the long term. In this way the memory cells capture the autoregressive structure of time-series [68].

Figure 3.9 demonstrates the structure of one LSTM memory cell.

The following sections describe the state and gates that LSTM uses.

3.4.1 Cell State

The cell state at the current time will update the previous cell state by applying two main operations: element-wise multiplication and element-wise addition. The input vector at time t is represented by x_t, while the hidden vector at the previous time is represented by h_{t-1}.

3.4.2 Forget Gate

The forget gate decides how much information to keep and to delete from the memory, and in this way LSTM avoids the exploding gradient problem. The

forget gate is computed as follows:

$$f_t = \sigma(W^{xf} x_t + W^{hf} h_{t-1} + b_f). \tag{3.27}$$

Here σ represents the activation function and b_f represents the bias vector.

3.4.3 Input Gate

The input gate is responsible for updating the cell state and controlling the input. The input gate is computed as follows:

$$i_t = \sigma(W^{xi} x_t + W^{hi} h_{t-1} + b_i). \tag{3.28}$$

Also, in this step, the input node should be computed as follows:

$$i_g = \tanh(W^{xg} x_t + W^{hg} h_{t-1} + b_g). \tag{3.29}$$

After computing the input gate and node, the cell state at time t will be updated as follows:

$$C_t = (C_{t-1} \odot f_t) + (i_t \odot g_t), \tag{3.30}$$

where b_i and b_g represent the bias vectors, and $\odot$ indicates element-wise multiplication.

3.4.4 Output Gate

The output gate is computed as follows:

$$o_t = \sigma(W^{xo} x_t + W^{ho} h_{t-1} + b_o). \tag{3.31}$$

In this case, b_o represents the bias vector. After computing the output gate and cell state, the hidden unit at time t is computed as follows:

$$h_t = o_t \odot \tanh(C_t). \tag{3.32}$$

The output of the cell state is multiplied by the output gate, which is a value between zero and one [57].

3.5 Accuracy Measures

To summarise the fitness of a regression model, statistical measures have to be used. These measures calculate the difference between the actual value y and predicted value $\hat{y}$. The difference between the predicted and actual values is called the error. In most of the models the smaller the error the higher the accuracy of the model. Five accuracy measures are discussed in the following sections.

3.5.1 Mean Squared Error

Mean squared error (MSE) is the average of the squares of individual errors. This measure produces a non-negative value and the highest accuracy is zero. MSE is given by:

$$MSE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.33}$$

for predicted value $\hat{y}_i \in \mathbb{R}, \forall i = 1,2,\ldots,n$ [30].

3.5.2 Mean Absolute Error

Danjuma [25] defined the mean absolute error (MAE) as the average of the absolute individual errors. The sign of the error is ignored to decrease the effect of outliers. This measure produces non-negative values. The highest MAE accuracy is zero. MAE is given by:

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{3.34}$$

where $\hat{y}_i \in \mathbb{R}, \forall i = 1,2,\ldots,n$ denotes the predicted value.

3.5.3 Median Absolute Error

The median absolute error (MedAE) produces a non-negative value and the highest accuracy is zero. The MedAE is robust to outliers, and is given by:

$$MedAE(y,\hat{y}) = Median(|y_i - \hat{y}_i|), \tag{3.35}$$

where $\hat{y}_i \in \mathbb{R}, \forall i = 1,2,\ldots,n$ denotes the predicted value.

3.5.4 Mean Absolute Percentage Error

The mean absolute percentage error (MAPE) is the average of absolute percentage errors and indicates the percentage of error in regression predictive models [37]. MAPE is defined as:

$$MAPE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{3.36}$$

where $\hat{y}_i \in \mathbb{R}, \forall i = 1,2,\ldots,n$ denotes the predicted value.

3.5.5 R^2 score

The R^2 score is a normalised version of MSE and is given by:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.37}$$

The highest R^2 accuracy is one. If the predictions are unrelated to the actual values then R^2 will be zero. The R^2 value can be negative for anti-symmetric functions.

3.6 Summary

This chapter presented the machine learning and accuracy measures that are used in this study. Variants of the ANN model are used, specifically MLP (presented in Section 3.2) and LSTM (presented in Section 3.4). The latter is an improved RNN model (Section 3.3) that solves the exploding and vanishing gradient problem (refer to Section 3.3.4). Section 3.5 presented a number of accuracy measures for regression and time-series data. The next chapter introduces some of the processes performed on the dataset for the study.

Chapter 4

Data Preprocessing

In this chapter, concepts like time-series data, stationarity, correlation, outlier detection, and outlier removal are discussed. The chapter also presents the water quality variables under study as well as the datasets used. The aim in this chapter is to prepare the dataset for the training process in Chapter 5.

4.1 Preprocessing Procedure

This section presents the steps that are applied to the dataset as preprocessing, to understand the underlying characteristics of the data.

- Take one water quality variable dataset (one time-series) at a time. If the series contains null values then replace them with the mean of the adjacent values.
- Find the correlation matrix between the variables and confirm the correlation degree.
- Apply outlier detection methods (Gaussian, isolation forests, k-means and one-class support vector machine), with a certain outlier fraction.
- Compare the results of the different outlier detection methods. An outlier is a data point that was classified as an outlier by all of the applied outlier detection methods.

4.2 Dataset

This study uses a number of water quality variables, collected at a water station in the United States called Hog Island (41.6423 N, 71.2800 W). The dataset was

obtained from the United States Geological Survey[1]. The start date of collection varies with different variables. All the details of the variables used in the study are available on the website for Hog Island[2] from October 2010 to the present. All the data is available to download[3].

Table 4.1 lists some of the water quality variables used in the study with their mean, standard deviation, minimum and maximum values, number of observations and unit. The variables differ in scale and the amount of data.

Table 4.1: Hog Island water quality variables and their statistical properties.

Variable	No. Obs	Days	Min	Max	Mean	Std	Unit
Specific conductance	573978	2391	30400	57700	45760.53	2238.26	microsiemens/cm
Dissolved oxygen	597881	2491	2.4	19	8.75	2.35	milligrams/liter
Chlorophyll	219990	916	0	130	13.18	12.13	mg/L
Turbidity	374380	1559	0	255	3.16	2.25	FNU
Temperature	603540	2514	-1.8	30.8	13.28	7.90	C°
Sampling depth	20922	87	1.09	11.4	6.45	1.71	feet
pH	403987	1683	7.1	8.7	7.87	0.22	moles/liter
Chlorophylls	358472	1493	0.1	376	14.81	15.14	micrograms/liter
Surface elevation	595159	2479	-4.67	10.89	1.32	1.71	feet
Tidal prediction	20922	87	-2.57	3.54	0.44	1.42	feet
Salinity	573978	2391	18.9	38.5	29.65	1.62	PSU
Difference obs-prd elv	17623	73	-2.88	3.52	0.56	0.91	feet

The number of observations for each variable depends on the start date of recording that particular variable. Variables like specific conductance and dissolved oxygen have a higher number of observations compared to turbidity and chlorophylls.

Figures 4.1, 4.2, 4.3 and 4.4 show the temperature, pH, dissolved oxygen and chlorophyll datasets over time.

The dynamics of the variables are different. Dissolved oxygen and temperature show periodic behaviour with some fluctuations in the dissolved oxygen, while pH and chlorophyll have dynamics that seem more random. Chlorophyll values are available for only three years (and some other variables too) which is less than other variables like pH and temperature.

The next sections introduce the nature of the dataset.

[1] https://www.usgs.gov/mission-areas/water-resources/data-tools
[2] https://waterdata.usgs.gov/ny/nwis/uv?site_no=01311143
[3] https://help.waterdata.usgs.gov/tutorials/overview/a-primer-on-downloading-data

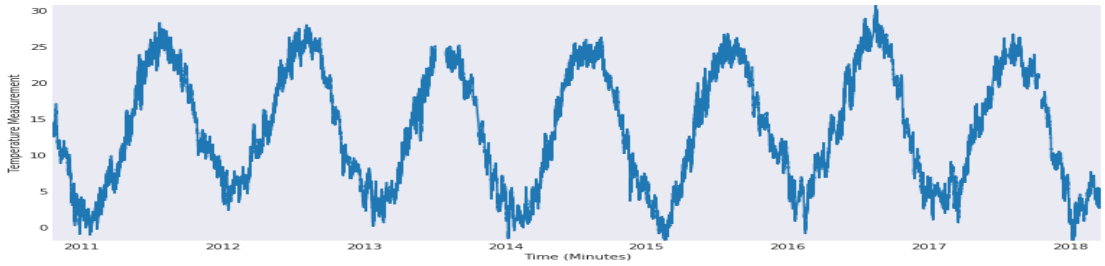


Figure 4.1: Temperature values over eight years collected in six-minute time intervals.

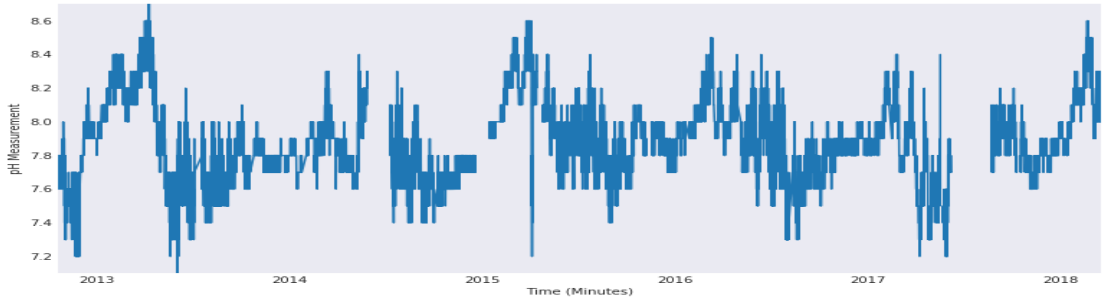


Figure 4.2: pH values over eight years collected in six-minute time intervals.

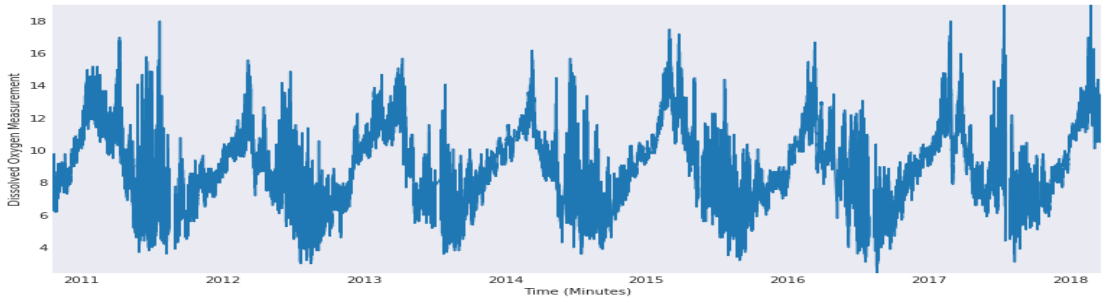


Figure 4.3: Dissolved oxygen values over eight years collected in six-minute time intervals.

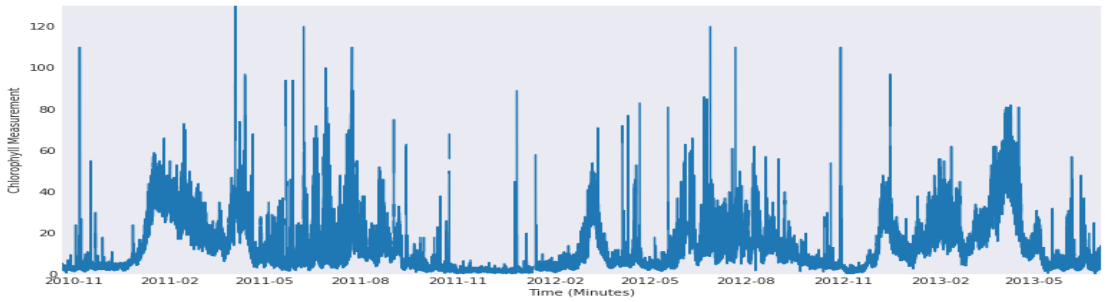


Figure 4.4: Chlorophyll values over three years collected in three-minute time intervals.

4.3 Time-Series Data

Time-series data (TSD) is a group of observations (data points) collected in constant time intervals. The mathematical representation of TSD is given by a set of vectors $x(t)$ where $t = 0,1,2,\ldots$ denotes time [8]. Time is an independent variable and the variable x is dependent on time. TSD is mostly used in forecasting future values [32]. It has to be handled differently than regression problems because of the temporal structure which makes the observation time-dependent. TSD also have some trends and seasonality features that may not exist in regression datasets.

4.3.1 Types of Time-Series Data

A multivariate TSD is a time-series that contains observations of more than one variable, while a univariate TSD contains only one variable's observations [8]. This study converts a multivariate time-series to univariate by considering only one variable at each experiment (see Figure 4.5). A univariate time-series has two dimensions, while a multivariate time-series has three dimensions.

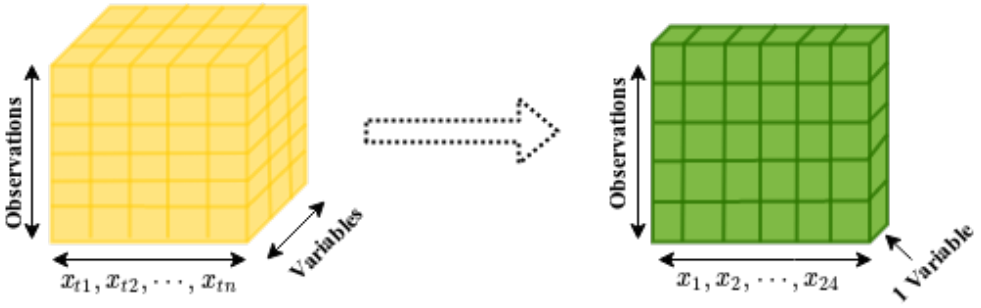


Figure 4.5: The multivariate and univariate representation of time-series data.

In Figure 4.5 the number of observations remains the same in both univariate and multivariate series (y axis), and the period for the observed values is also the same (denoted by $x_1, x_2, \ldots, x_{24}$ in the x axis). However, the multivariate series has a third dimension representing the different variables (z axis), while a univariate series has only one variable. Figure 4.6 presents specific conductance data. This series is a univariate TSD since only one variable is recorded over time.

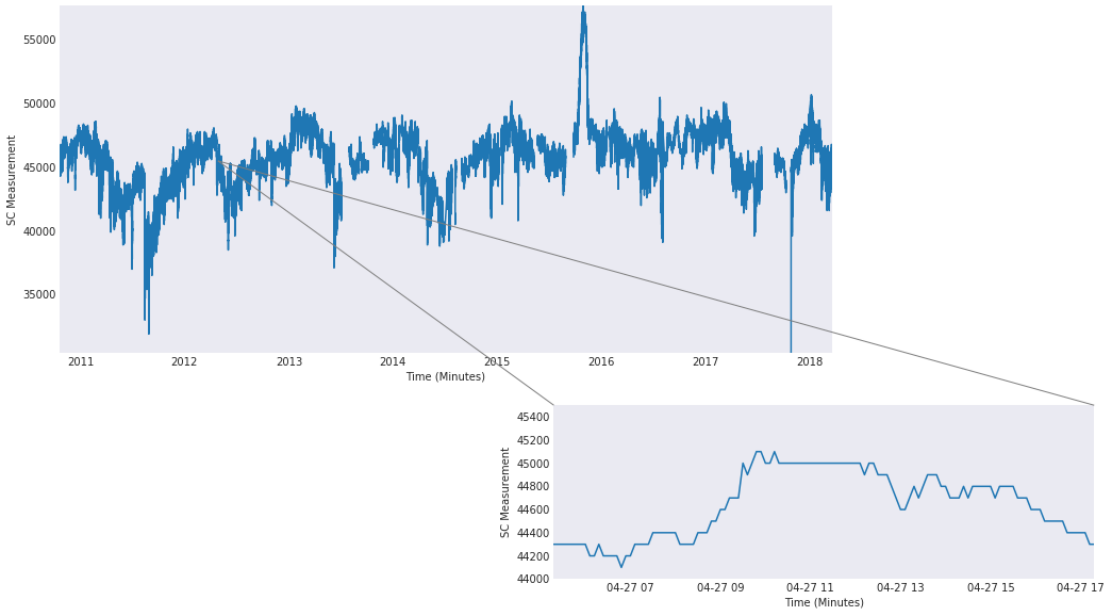


Figure 4.6: Specific conductance values over eight years collected in six-minute time intervals, with a zoom-in of the second year (2011).

4.3.2 Time-Series Data Components

Time-series is affected mainly by four components: trend, cyclical, seasonal and irregular or residual components. The definitions below were obtained from [1] and [8].

- Trend: the tendency of the data to increase or decrease over long periods. If the data shows an increasing behaviour over time then it has a positive trend. Conversely, if the data shows a decreasing behaviour over time then it has a negative trend.

- Seasonality: variations in the data that appear at regular intervals. In Figure 4.3, the dissolved oxygen series has a seasonality property. It reaches the same value in a certain period of the year.
- Cyclic: the movement of the data in a cyclic period. Examples of this can be seen in Figures 4.1 and 4.3 which represent dissolved oxygen and temperature series.
- Residuals: a sudden change in the data. Figure 4.4 contains some residuals, during April 2011 among others.

Identifying these components in the series might help in determining the type of preprocessing techniques that should be applied to the data. Moreover, finding residuals may assist in determining and removing outliers in the data.

4.4 Stationarity

Time-series data is defined to be stationary if its statistical distribution does not change over time. The mean, variance and autocorrelation remain the same. No trends, seasonality or occur over time. Stationarity is assumed in many statistical procedures for time-series analysis, therefore it can be effective to transform the data if it is not stationary. Stationarity also helps in identifying driving factors. To find the correlation between two time-series data (two water quality variables), seasonality and trend have to be removed from the datasets.

4.4.1 Stationarity Types

There are three main types of stationarity: strict, trend and difference. The definition of each one is explained below.

- Strict stationarity: this series has constant statistical properties like mean, variance and covariance.
- Trend stationarity: this series does not have unit root (defined in Section 4.4.3) but it has a trend. It can be converted to strict stationarity by removing the trend.
- Difference stationarity: a new series obtained by taking the difference between consecutive observations in the series.

4.4.2 Importance of Stationarity

The main reasons to apply stationarity processes to time-series data are as follows.

- It is necessary for forecasting models because they assume stationarity of the series. If the TSD has constant statistical properties over time then it will likely follow the same properties in the future [32].
- Stationary TSD is easier to implement compared to non-stationary TSD.

4.4.3 Testing Stationarity

The stationarity of a series can be checked (in an approximate sense) by simply looking at the graph to see if there is trend or seasonality. However, one can test if a series is stationary by several methods. This includes unit root methods like the augmented Dickey-Fuller (ADF) and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [7]. Unit root tests indicate that the statistical properties of a series are not constant with time. A sequence of time-series data can be defined as:

$$Y_t = \alpha Y_{t-1} + \epsilon_t, \tag{4.1}$$

where Y_t and Y_{t-1} are the values of the series at time t and $t-1$ respectively. ϵ_t is the error term at time t, and α is a coefficient. The value of the series at $t-1$ is defined as follows:

$$Y_{t-1} = \alpha Y_{t-2} + \epsilon_{t-1}. \tag{4.2}$$

Therefore, the value of Y_t will be the accumulated value of all values prior to Y_t. If m values are present then Y_t will be:

$$Y_t = \alpha^n Y_{t-n} + \sum_{i=1}^{n} \alpha^i \epsilon_{t-i}. \tag{4.3}$$

If the value of α is 1 (unit root), then the variance will be a function of time (increase with time). The unit root tests if the series has 1 as a value for α [4].

4.4.3.1 Augmented Dickey-Fuller

The ADF test is one of the unit root methods. It tests two hypotheses of difference stationarity [33]:

- null hypothesis: the series has a unit root, hence, it is non-stationary;
- alternate hypothesis: this hypothesis rejects the null hypothesis and suggests the series is stationary.

Given a time-series $x_1, x_2, \ldots, x_n$ with the following autoregressive model at time t:

$$\Delta x = x_t - x_{t-1} = (\phi - 1)x_{t-1} + \varepsilon_t, \tag{4.4}$$

ϕ represents the coefficient of the first lag on x. If it is one then it is a unit root, which agrees with the null hypothesis of this test.

The test produces a test statistics value that will be compared to a threshold value (a confidence interval value). If the test statistics value is greater than the threshold then it fails to reject the null hypothesis and the series is not stationary, otherwise the series is stationary. A lower value suggests rejecting the null hypothesis and thus that the series is stationary.

The function adfuller in the StatsModels library for Python implements the test [33]. As an example, the function is used to test a specific conductance time-series for stationarity. Results are listed below.

ADFStatistic	-4.146027
p-value	0.000812
Critical values:	
1%	-3.431
5%	-2.862
10%	-2.567

The results above suggest rejecting the null hypothesis with a significance level of less than 1%, since -4.146027 is less than -3.431. Therefore, the specific conductance series is stationary.

4.4.3.2 Kwiatkowski-Phillips-Schmidt-Shin

KPSS is another method to test trend stationarity of a series. It has the opposite hypothesis to the ADF test. If the test statistics value is greater than the threshold, then the null hypothesis should be rejected and that implies a non-stationary series. Results of this test on the specific conductance series are listed below.

KPSSStatistic	3.643854
p-value	0.010000
Critical values:	
1%	0.739
5%	0.463
10%	0.347

The results above failed to reject the null hypothesis. The specific conductance statistics value is higher than all confidence intervals. Therefore, it suggests that the series is not trend stationary.

To check if all water quality variable series are stationary both test were applied. Refer to Table 4.2 for the suggested method to convert a series to stationary.

Table 4.2: Suggested methods to convert non-stationary series to stationary ones.

ADF	**KPSS**	**Final result**	**Suggested solution**
1 [1]	1	strictly stationary	-
1	0 [2]	difference stationary	apply differencing
0	1	trend stationary	remove trend
0	0	non-stationary	apply differencing and remove trend

[1] 1 refers to stationary series.
[2] 0 refers to non-stationary series.

4.4.4 Conversion to a Stationary Time-Series

This section covers techniques used in converting non-stationary series into stationary series.

4.4.4.1 Differencing

Differencing is the process of creating a new series out of the difference of every two consecutive observations in a given series. This process helps in obtaining a constant mean. Mathematically, differencing is expressed as follows:

$$Z_t = Y_t - Y_{t-1}. \tag{4.5}$$

From the results in Section 4.4.3.1, specific conductance was found to be non-stationary in terms of difference. Therefore, the difference process has to be

applied. Figures 4.7 and 4.8 illustrate the difference between stationary and non-stationary specific conductance series.

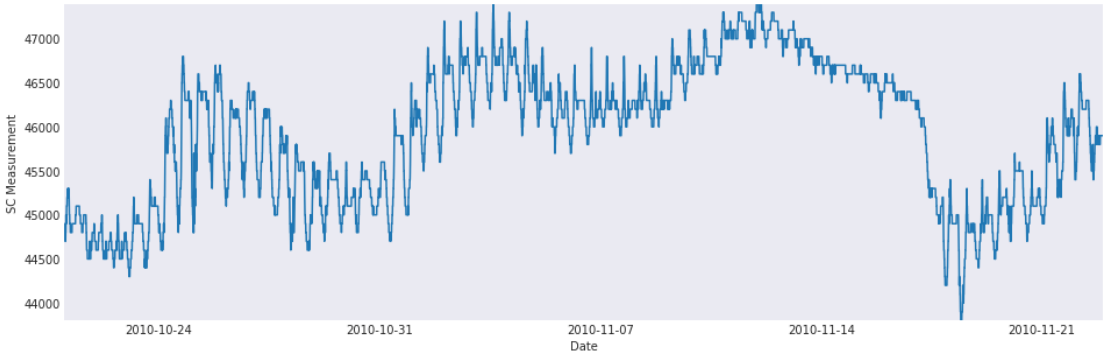


Figure 4.7: Non-stationary specific conductance series.

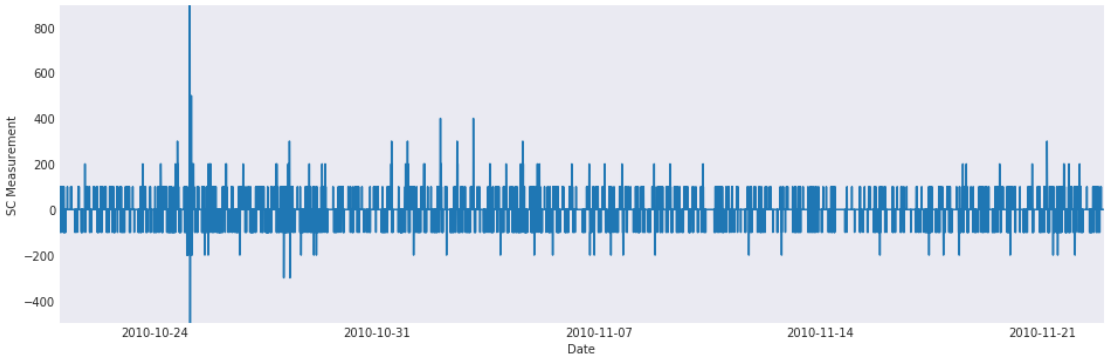


Figure 4.8: Stationary specific conductance series. Differencing was applied to obtain this series.

The differencing stationarity can also be explained using the figures. If Figure 4.7 is split into four regions, then every region will have a different range of values. If Figure 4.8 is split into the same four regions then all the values will fall under the same range except for some residuals.

4.4.4.2 Seasonal Differencing

Seasonal differencing is the process of calculating the difference between two observations; one in the current season and one in the previous season. For example, differencing may be performed on the specific conductance values in

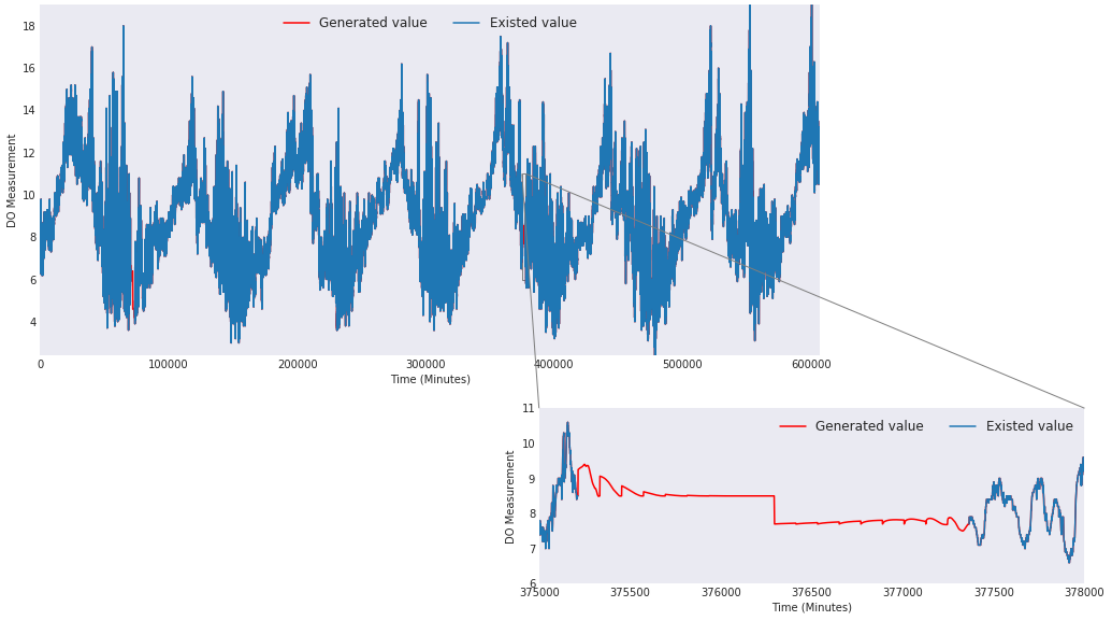


Figure 4.9: Dissolved oxygen measurements after filling gaps in the data.

January 2010 and the values in January 2011. This process may remove the trend from the series. In the equation below, Y_t is the current observation and Y_{t-n} is the observation that is n steps away from the current one:

$$Z_t = Y_t - Y_{t-n}. \tag{4.6}$$

4.5 Filling Gaps in Time-Series

The water quality dataset may contain missing or undefined values, denoted by NaN (not a number). It is important to fill gaps in TSD since it improves the accuracy of the prediction models [36].

Looking at the dataset, some indices contain NaN values. Such indices are replaced with the mean value of adjacent values. Some classical statistical and mathematical methods can also be used for filling gaps, by substituting missing values with the mean, nearest neighbour, linear, or cubic interpolation, among others [36], [21]. The mean of adjacent values method has been performed on the water quality series. The results obtained after filling gaps in the dissolved oxygen time-series are shown in Figure 4.9. The generated values from applying the adjacent mean method are highly dependent on the observations neighbouring the missing values. If there are multiple missing indices in a row, the rolling window process can be applied. It applies the adjacent mean method multiple

Table 4.3: Hog Island water quality variables ordered by their percentage of missing observations.

Variable	Percentage of missing values
Diff obs/pred elev	97.11
Temperature	96.72
Elevation/tidal prediction	96.58
Nitrate	86.49
Chlorophyll	65.50
Chlorophylls	43.78
Turbidity	41.28
pH	36.61
Salinity	9.99
Specific conductance	9.98
Dissolved oxygen	6.23
Sampling depth	5.32
Elevation	1.67

times until all the gaps between two observations are filled. The rolling window ensures completeness of the gap-filling process.

Table 4.3 shows the Hog Island water quality variables and their percentage of missing values. The difference in observed elevation variable has the highest percentage of missing values, while the elevation variable has the lowest.

In Figure 4.10, the percentage of missing values from the Hog Island water quality variables are shown in the form of bars. The higher the bar is, the more observations are missing in the series. Temperature, nitrate, elevation and tidal prediction have more than 80% missing values, while chlorophyll, chlorophylls, turbidity and pH have missing value percentages between 30% to 70%. Salinity, specific conductance, dissolved oxygen, sampling depth and elevation have less than 10% missing values.

4.6 Correlation Between Variables

It is important to identify water quality variables and potential relationships with themselves earlier in time (autocorrelation) and others (correlation). Identifying the degree of autocorrelation helps in determining the number of prediction steps in the future.

Figure 4.11 illustrates the correlation maps for some of the variables under

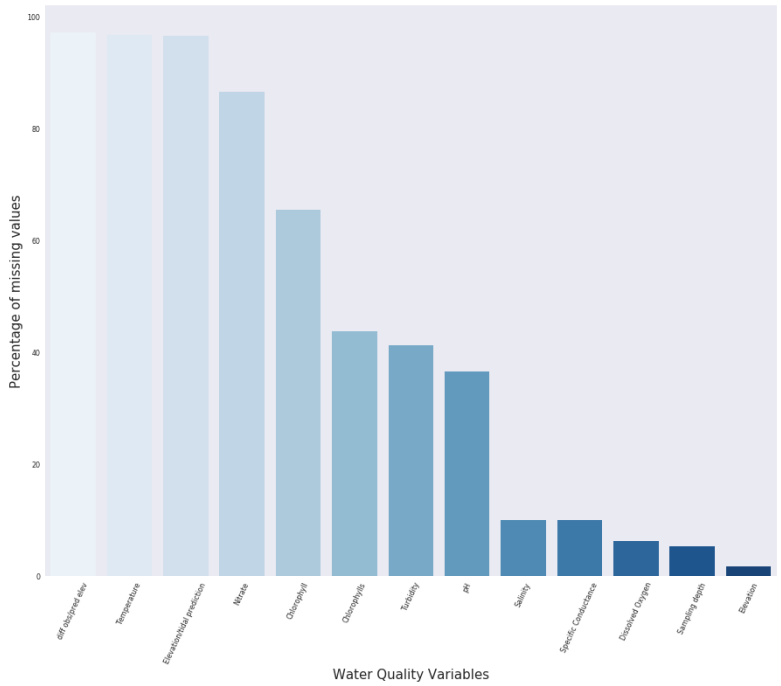


Figure 4.10: Missing data percentage of Hog Island water quality variables.

study.

The darker the square between a pair of variables, the more correlated they are. For example, dissolved oxygen is strongly positively correlated to chlorophyll. This finding confirms that the existence of algae in the water results in increasing the oxygen level due to the photosynthesis process. Also, sampling depth is strongly negatively correlated to dissolved oxygen. The level of oxygen in the surface of the water source is high because the water at the surface can mix with the atmospheric oxygen, and the level of oxygen decreases as one gets deeper into the water source.

4.7 Outlier Detection in Time-Series Data

Outlier detection is the process of identifying or classifying deviating observations from the normal ones [12]. Outliers in the datasets can occur due to human error or a defect in one of the components in a system [64], or problems in measurement devices [21]. Detecting outliers in datasets is important because of the misinformation that they may convey. It has been studied in various domains, and each domain has techniques that examine anomalies in data across time. There are several statistical methods proposed in the literature for outlier

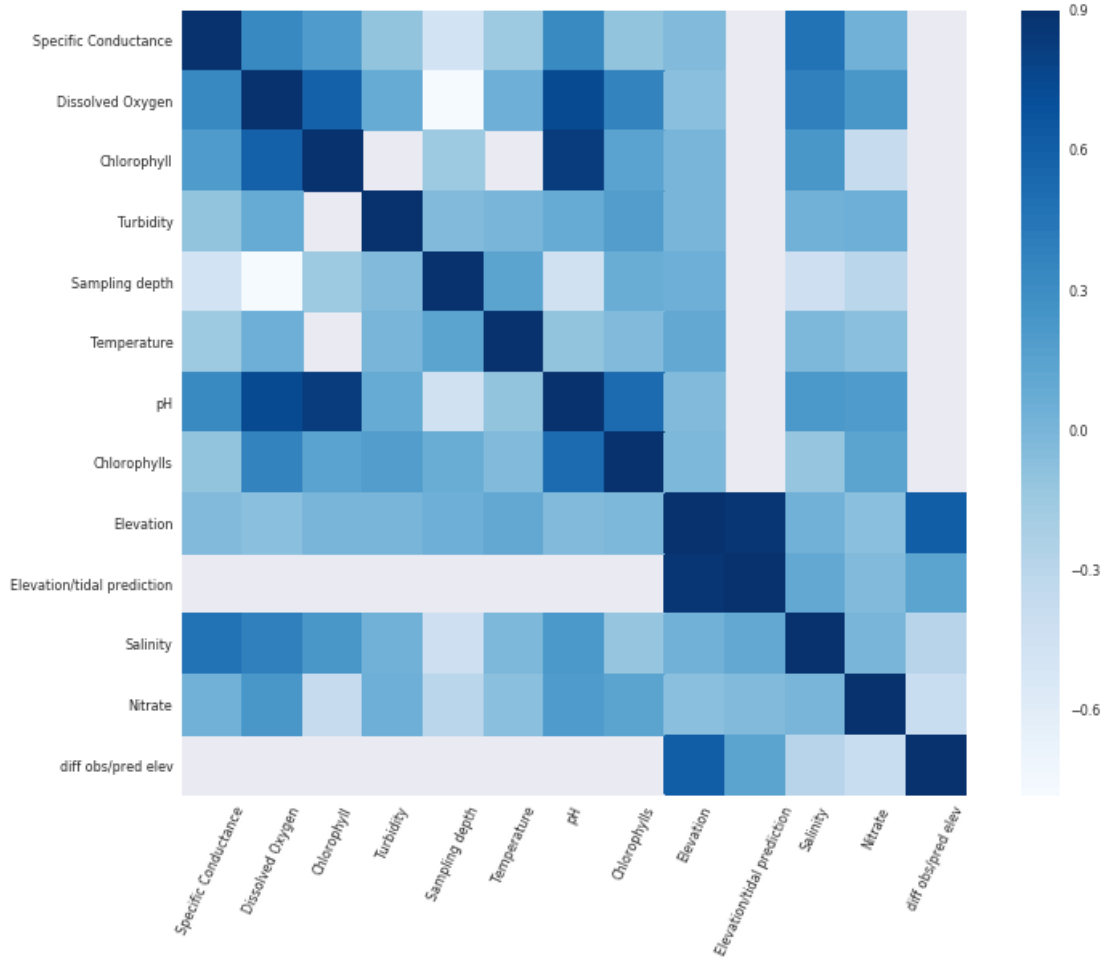


Figure 4.11: Correlation map for the Hog Island water quality variables.

detection, including autoregressive moving average, autoregressive integrated moving average, vector autoregression, cumulative sum statistics, and exponentially weighted moving average. Some studies also apply density-based, neural network-based or kernel-based machine learning methods [12]. However, none of the studies that were found apply ensemble methods to identify outliers. In most cases, different methods identify different outliers. In ensemble methods, all or most of the outlier detection methods have to agree on a point to be classified as an outlier. This study applies an ensemble method using four algorithms.

Figure 4.12 illustrates the flow of steps to identify outliers in a given time-series dataset. Section 4.7.1 covers types of the outliers and Section 4.7.2 presents methods for detecting outliers.

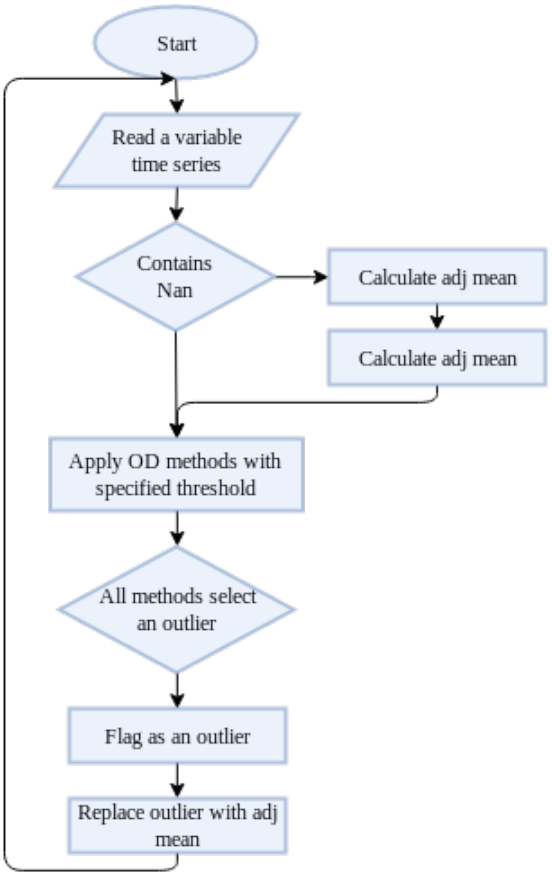


Figure 4.12: The steps performed to identify outliers in this study.

All of the previous processes mentioned in this chapter were applied before the step of detecting outliers. To identify an outlier, an ensemble of four methods will be applied.

4.7.1 Outlier Types

This section presents the main types of outliers that can be found in TSD.

4.7.1.1 Point Outliers

The main goal of the technique is to identify an outlier such that the removal of that point from the time-series results in a sequence that can be represented more accurately than the original one. Prediction models can be employed to do so.

4.7.1.2 Subsequence Outliers

A subsequence of a TSD that begins at a position is said to be an outlier if it has the largest distance to its nearest non-overlapping match.

4.7.2 Outlier Detection Methods

This section presents some of the methods used for detecting outliers in TSD. All methods in this section were applied to the Hog Island water quality variables.

Using several methods to detect outliers results in different points that may intersect or not. Figure 4.15 presents the difference in the results of the detection methods.

4.7.2.1 Isolation Forest

An isolation forest (IF) is an ensemble of isolation trees: $IF = \{t_1, t_2, \ldots, t_T\}$. Every isolation tree is constructed as described in Algorithm 3.

Algorithm 3: Isolation Tree Algorithm

input : D-dimensional training set X = $\{x_1, x_2, \ldots, x_N\}$
parameter : the lower dimension d, where $d \leq D$

1 $t = \emptyset$ (empty tree at the beginning);
2 **while** $(nrow((X) \neq 1)$ **do**
3 Select feature x_i of X randomly;
4 Select split point $p \in (min(x_i), max(x_i))$;
5 Add the node $N_{x_i,p}$ to t ;
6 Define matrix X_l containing samples where $x_i > p$;
7 Define matrix X_r containing samples where $x_i < p$;
8 $X = X_l$, go to 1;
9 Link the obtained tree as the left child of t;
10 $X = X_r$, go to 1;
11 Link the obtained tree as the right child of t;
12 return t;

IF can be used in identifying outliers in a number of steps. The average number of steps required to isolate a sample x in tree t with iterations $h_t(x)$ is given by:

$$h(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x). \tag{4.7}$$

IF assigns an anomaly score to every observation in the dataset. The anomaly score is defined as:

$$s(x,n) = 2^{-\frac{h(x)}{c(n)}}, \tag{4.8}$$

where $c(n)$ is a normalisation factor and n is the total number of samples. It is the average number of steps required to isolate a sample from the other samples in the dataset. This number is influenced by the number of samples n in the dataset. Therefore, $c(n)$ has different values based on that:

$$c(n) = \begin{cases} 2H(n-1) - 2(n-1)/n, & \text{if } n > 2, \\ 1, & \text{if } n = 2, \\ 0, & \text{if } n < 2, \end{cases} \tag{4.9}$$

where H is a harmonic number estimated as: $H(i) \approx \ln(i) + 0.5772156649$ [45]. Figure 4.15a shows the results of applying IF on specific conductance.

4.7.2.2 k-means

k-means is an unsupervised machine learning method that is used to cluster a dataset into k clusters. Clustering means dividing a large dataset into small datasets called clusters. Clustering by the method of k-means is done based on the distance from centroid points [17].

k-means can also be used to identify outliers. Algorithm 4 provides all the steps to obtain clusters using k number of clusters [2].

The value of k has significance on the performance of the method. To choose a suitable number of clusters for the dataset, the elbow curve may be used. It is a method that runs k-means on the dataset for a range of k values, and the goal is to validate the number of clusters. This method provides a graph of the explained variance score over values of k. The method starts with one cluster, calculates its score, and keeps increasing the number of clusters until it reaches a point where adding more clusters will not add more information. The goal of the elbow curve method is to choose the lowest possible k with a relatively high score [17].

k-means starts by choosing random centroid points for k clusters. All the points in the dataset are assigned to nearest clusters after finding the Euclidean distance between the observation position coordinates and centroid coordinates. The centroid points are then recalculated. This step executes iteratively until convergence or until it reaches a maximum number of iterations. In this way, k clusters will be formed. A cluster number is assigned to every point in the

dataset. The distances of all points to their cluster centroids are compared to a predefined threshold. If an observation's distance to its cluster centroid is larger than the threshold, then it is considered as an outlier.

In the Hog Island dataset, a range between 1 and 20 is used to find the best k value. Figure 4.13 shows that the best number of clusters that can be used is 10. k-means with 10 clusters were applied to obtain the results in Figure 4.14a.

Algorithm 4: k-means Algorithm

```
   input     : d-dimensional training set X = {x_1, x_2, ..., x_N}
   parameter : The number of clusters k, where k > 0
 1 Initialise k random centroids;
 2 while (method has not converged) do
 3     for x_i ∈ X do
 4         shortest = 0;
 5         membership = null;
 6         for centroid c do
 7             dist ← distance(c);
 8             if dist < shortest then
 9                 shortest ← dist;
10                 membership ← c;
11     Go to 2 to recalculate all the centroids;
12 Return clustered dataset;
```

Applying k-means on the specific conductance series resulted in detecting the outliers presented in Figure 4.15b.

4.7.2.3 Gaussian Distribution

The Gaussian distribution (GD) is also called the normal distribution. It is used in probabilistic machine learning when the distribution of the observations forms a bell curve shape.

If the distribution has zero mean and unit standard deviation then it is called the standard normal distribution. Its probability density function is defined as:

$$p(x) = \mathcal{N}(x;0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}. \tag{4.10}$$

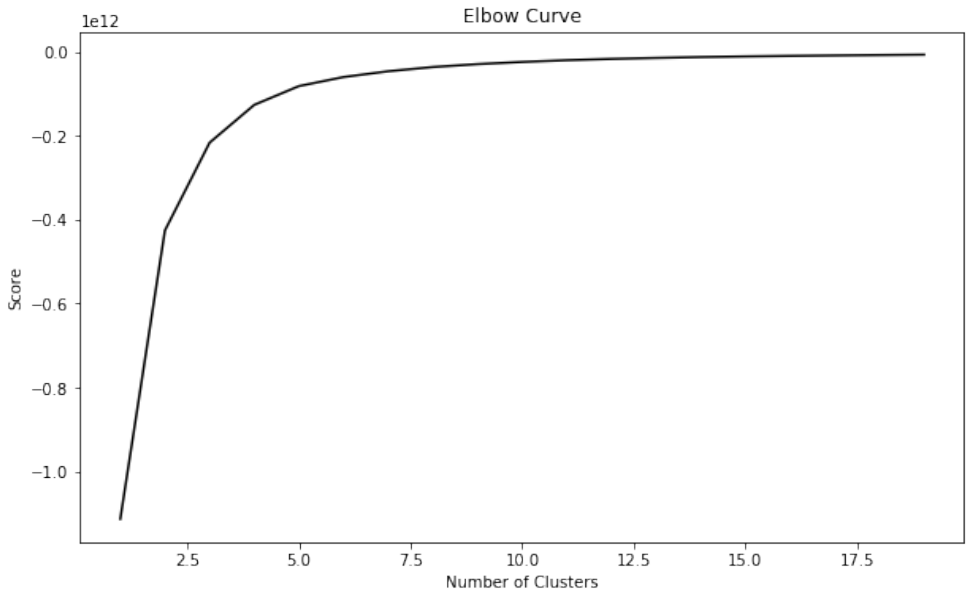
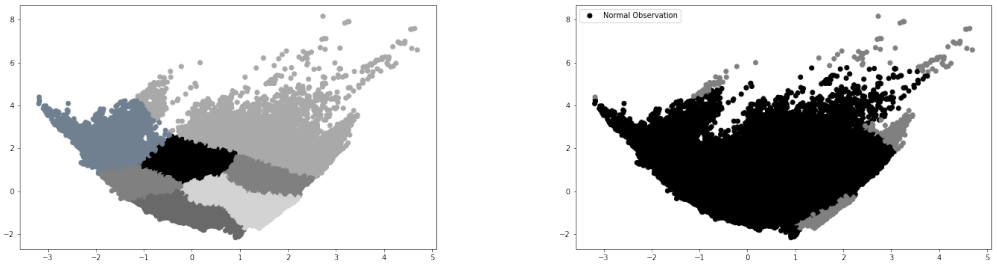


Figure 4.13: The elbow curve shows the results of applying k-means clustering for a range of k values from 1 to 20.



(a) k-means result with $k = 10$.

(b) k-means result with $k = 2$.

Figure 4.14: The results of applying k-means clustering. Ten and two clasess are presented. The latter demonstrates the normal and outlier observations.

GD is a statistical method to test for the existence of outliers in datasets. This method assumes the dataset follows a Gaussian distribution and it finds the mean value and variance for all features in the dataset. Using these values, a probability density function can be obtained. GD was applied to detect outliers in the specific conductance series, as shown in Figure 4.15c. Algorithm 5 describes the steps of finding outliers in a dataset.

Algorithm 5: Gaussian Distribution Algorithm

input : d-dimensional training set $X = \{x_1, x_2, \ldots, x_N\}$

for *feature:* $i \in \{1, \ldots, D\}$ **do**
  $\mu_j \leftarrow \frac{1}{N}\sum_{i=1}^{N} x_j^i$;
  $\sigma_j^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N} (x_j^i - \mu_j)^2$;
  if *observation* < *threshold* **then**
    shortest $\leftarrow$ dist;
    membership $\leftarrow c$;

4.7.2.4 One-Class Support Vector Machine

The support vector machine (SVM) is a kernel-based method that is used in solving classification problems, by mapping the observations from its dimension (input space) to a higher dimension (feature space) using a nonlinear kernel function. The goal of SVM is to find the optimal hyperplane in the feature space that has the maximum margin between classes [12].

The one-class support vector machine (OCSVM) is a binary classifier. It has proven successful for one-class classification problems [12], and computes a function such that most of the observations are in the region where the function is non-zero. OCSVM transfers observations to a higher dimension and formulates a hyperplane that separates observations from the origin with the maximum margin. In this case, the origin point is treated as an outlier. OCSVM classifies new observations by testing if they belong to a defined class (normal observations) or not. Separating observations from the origin point involves solving the following quadratic minimisation problem [61]:

$$\begin{aligned} &\min_{w,\xi_i,\rho} \frac{1}{2}||w||^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \\ &\text{s.t. } (w.\phi(x_i)) \geq \rho - \xi_i, \forall i = 1, \ldots, n \\ &\xi_i \geq 0, \forall i = 1, \ldots, n \end{aligned} \tag{4.11}$$

The distance to the origin in feature space is presented by ρ. The variable w is the parametrisation of the hyperplane to separate the origin from the observations. ξ is a slack variable that introduced to allow some points to lie within the margin which results in soft margin. ν is the smoothness parameter which is the proportion of outliers expected in the data. The results of applying OCSVM on the specific conductance series are shown in Figure 4.15d.

4.8 Data Scaling

Data scaling is a data transformation technique that is applied to the datasets. It involves changing the values of the dataset to fall within a specific range such as zero to one. Changing the scale of the dataset to a specific range may result in increased model accuracy [62]. Also, scaling the training set (refer to Section 4.9) accelerates the learning process. There are different techniques for scaling the dataset. Normalisation is a type of scaling where all the values fall exactly between zero and one. Min-max, z-score and decimal scaling normalisation are the most common normalisation techniques [48].

In this study, the min-max normalisation technique was used to transform the water quality time-series to range between zero and one, using the following formula:

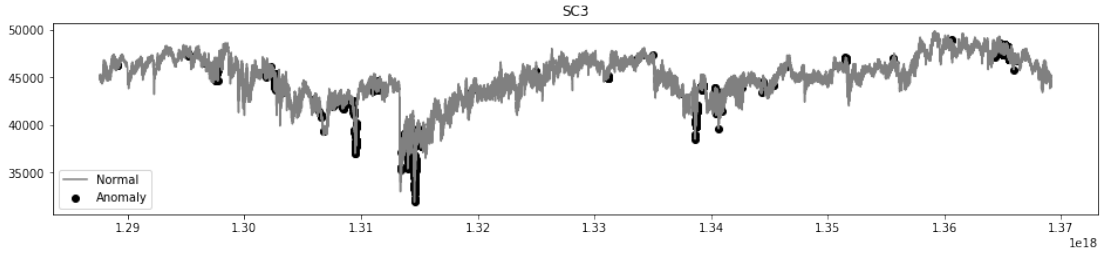
$$\bar{x_t} = \frac{x_t - x_{min}}{x_{max} - x_{min}}, \tag{4.12}$$

where $\bar{x_t}$ is the scaled value, x_t is the original value, and x_{min} and x_{max} are the minimum and maximum values in the dataset, respectively. Scaling the dataset makes it ready for model training and testing. After scaling the water quality variables, an inverse transformation should be applied to return the scaled values to their original values.

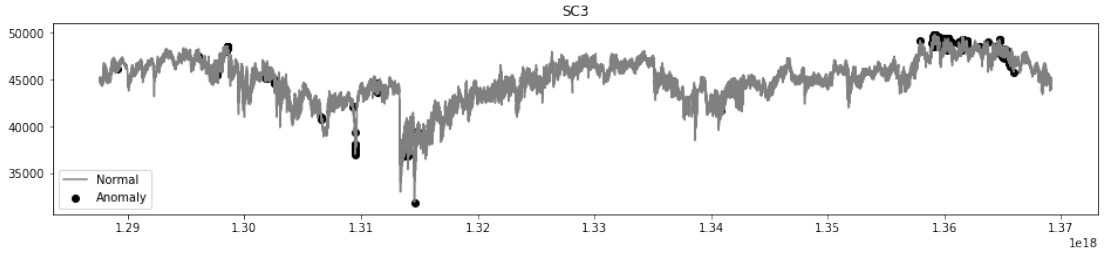
4.9 Data Split

Data splitting is the process of partitioning the data into disjoint portions [59]. Machine learning aims to build models that can generalise for new examples, not seen during training. A splitting strategy is necessary for model selection, training and evaluation. A common technique to split the dataset is the holdout method [60], where data is split into training, validation and testing datasets.

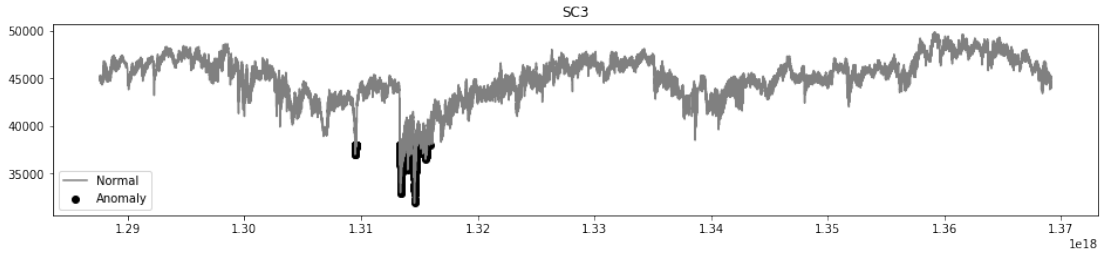
- The training dataset is the portion of the data used in the training process. Supervised learning applications consist of training and testing phases. The training phase analyses the training set to minimise the loss function.



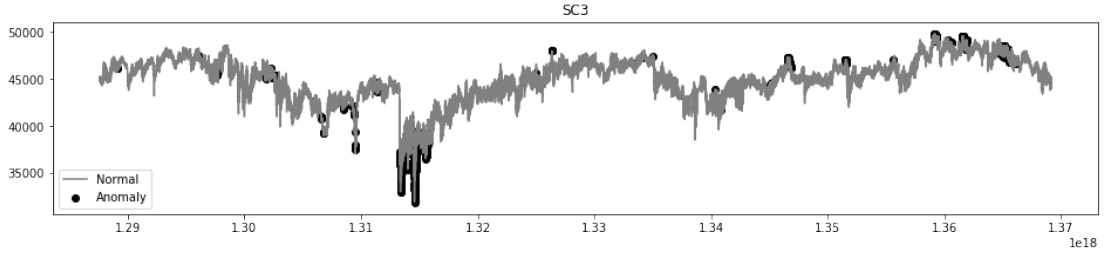
(a) outliers and normal observations discriminated using IF.



(b) outliers and normal observations discriminated using k-means.



(c) outliers and normal observations discriminated using extreme values in the Gaussian distribution.



(d) outliers and normal observations discriminated using OCSVM.

Figure 4.15: Outliers and normal observations of specific conductance obtained after applying IF, k-means, GD and OCSVM. These results used an outlier fraction of 1%.

- The validation dataset is the portion of the data used for performance evaluation during the training phase. The main aim of this dataset is to check when the model should stop learning to avoid overfitting.
- The testing dataset is the portion of the data that is used in predicting the output for observations not seen during training. This dataset is used in the testing phase and works as a confirmation for the model's ability to generalise.

In this study, the data was split 60:20:20 for training, validation and testing respectively. Figure 4.16 demonstrates the split on the temperature time-series.

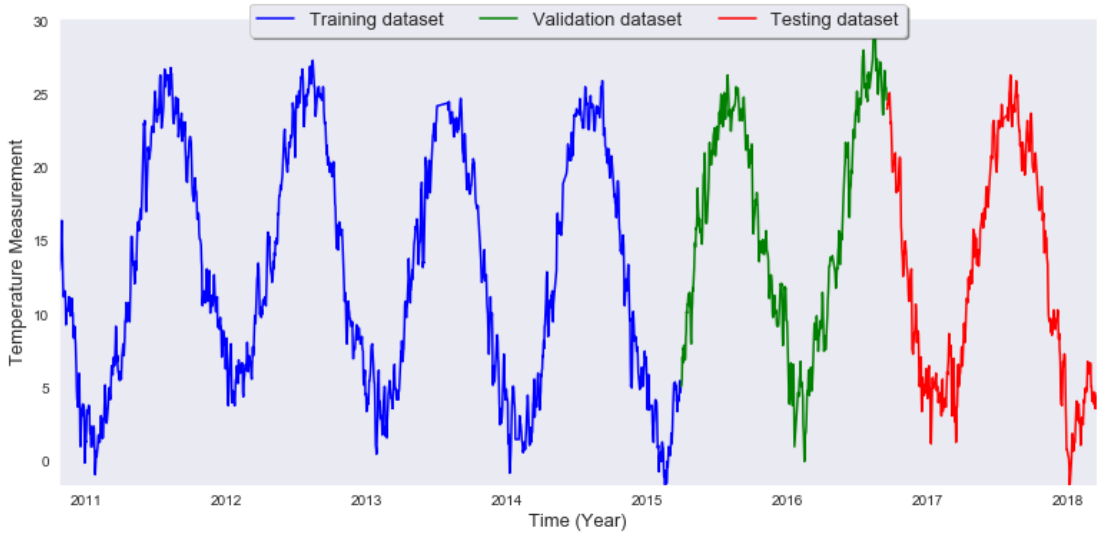


Figure 4.16: The temperature series split into training, validation and testing sets.

The sets are differentiated by colour. The first set which is bigger represents the training set, followed by the validation set, and lastly, the testing set. The last two sets are equal in size but they might have different fluctuations.

4.10 Summary

This chapter attempted to understand the primary features of water quality variables in the Hog Island dataset. Time-series data was introduced in Section 4.3 and it has been found that the Hog Island dataset is a multivariate time-series dataset. The stationarity property and its importance were introduced in

Section 4.4. Section 4.5 presented the gap filling process using average mean value, and Section 4.6 showed the correlation between the water quality variables in the Hog Island dataset. Section 4.7 covered the ensemble model used in detecting outliers, which combined isolation forests, k-means, Gaussian distribution, as well as one-class support vector machines. Lastly, Section 4.8 and Section 4.9 discussed the data scaling process and data splitting, respectively. The next chapter presents and discusses the results obtained after applying the models on the processed dataset.

Chapter 5

Results and Discussion

The study aims to compare the performance of multilayer perceptron (MLP) and long short-term memory (LSTM) models on predicting the values of water quality variables (discussed in Section 1.4). This chapter presents the models used in the prediction of the variables. The results obtained by applying these models will be discussed. The same input settings were used to compare the accuracy of the models. The input values used in the MLP and LSTM models are the dataset after applying outlier removal and gap filling, as discussed in Chapter 4. Section 5.1 of this chapter gives the hyperparameters used in MLP and LSTM. Section 5.2 compares the predicted values obtained with both models.

The chapter also illustrates the loss values obtained by mean squared error (MSE). Furthermore, Section 5.4 summarises the accuracy results for the experiments. Lastly, Section 5.5 discusses the obtained results.

5.1 Model Architectures

The MLP and LSTM models were structured differently to obtain the results. Different structures were tested and the hyperparameters in Table 5.1 were the best across all trials.

All the models were implemented in Python using the Keras, Pandas, Matplotlib and NumPy libraries[1]. Figure 5.1 illustrates the stacked structure of the layers used in building the MLP and LSTM models.

[1] https://keras.io, https://pandas.pydata.org, https://matplotlib.org, https://numpy.org

Table 5.1: Hyperparameters used in the models.

Parameter	MLP	LSTM
Learning rate	0.01	0.001
Number of layers	3	3
Input units	128, 64, out[1]	in[2], 128, 64, out
Dropout layers	1	2 (with 50%)
Activation function	Linear	Linear
Optimiser	SGD	Adam
Batch size	512	512
Iteration	100	100

[1] Number of hours or days to forecast.
[2] Number of input hours or days.

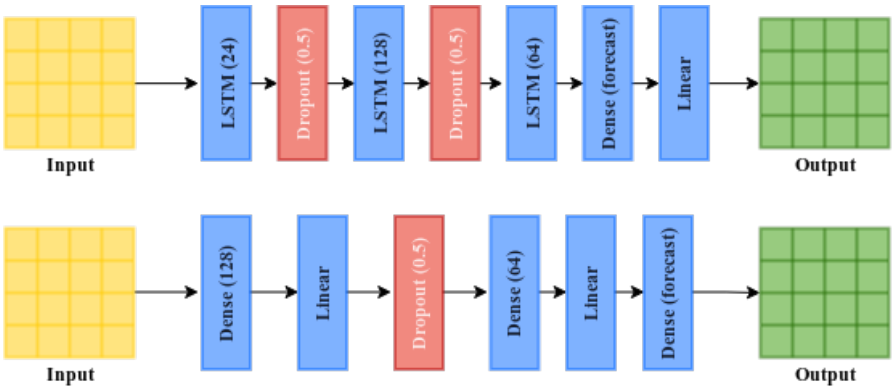


Figure 5.1: The architectures of the MLP (bottom) and LSTM (top) models used in the prediction process.

Some hyperparameters are shared between MLP and LSTM; for example, the number of layers in both of the models are three, they both use linear as an activation function, and the batch size and the number of iterations are set the same for both models. The number of dropout layers and optimiser, as well as the input units, are different.

5.2 Visualisation of Results

The same inputs for the two models were set, which include scaled time-series of the water quality variables at Hog Island station such as specific conductance,

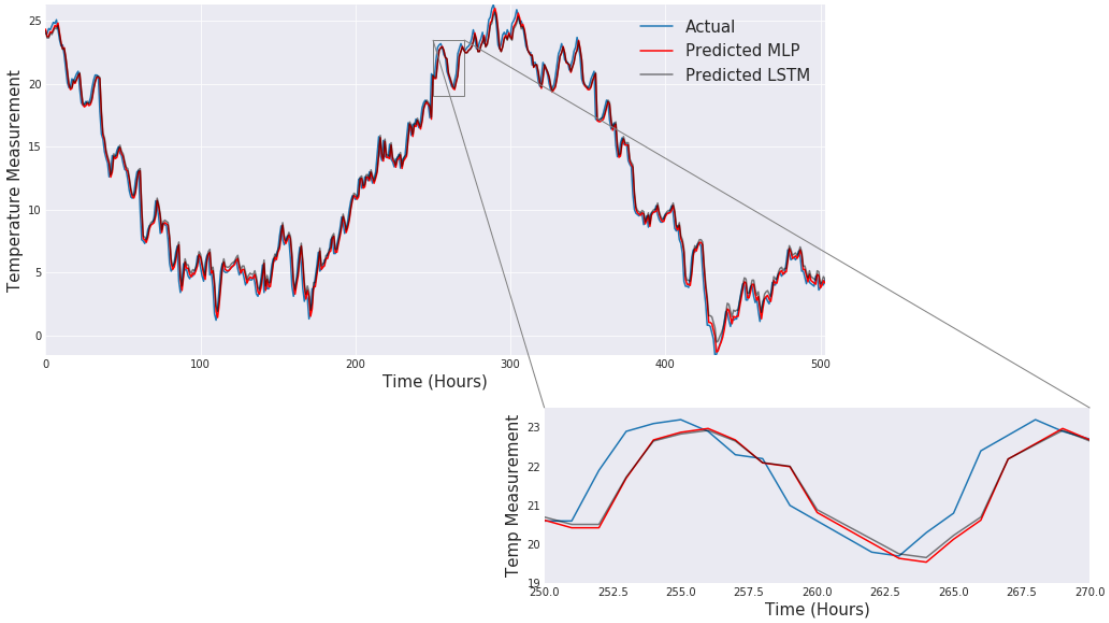


Figure 5.2: Day-to-day temperature prediction, using one hour values.

dissolved oxygen and turbidity. The models were trained using the parameters discussed in Section 5.1 to predict the following:

- the following day's measurement (daily value);
- the following day's measurements (hourly values);
- measurements one week into the future (hourly values);
- measurements one month into the future (hourly values).

This section presents some of the results obtained after applying MLP and LSTM to the dataset. The figures display the predicted values by both models, and compare them to the actual values, on the hold-out test set. Figure 5.2 predicts the next day's temperature value using the previous day's temperature value. Figure 5.3 shows results from MLP and LSTM that take the daily 24-hour temperature values as input and predict the following day's 24 temperature values as output. Figure 5.4 takes the daily 24-hour temperature values as input and predicts the 24 temperature values one week ahead as output. Figure 5.5 takes the daily 24-hour temperature values as input and predicts the 24 temperature values one month ahead as output.

Figure 5.2 indicates that the predicted temperature values by MLP and LSTM are close to the actual values, where one day's value is considered to predict

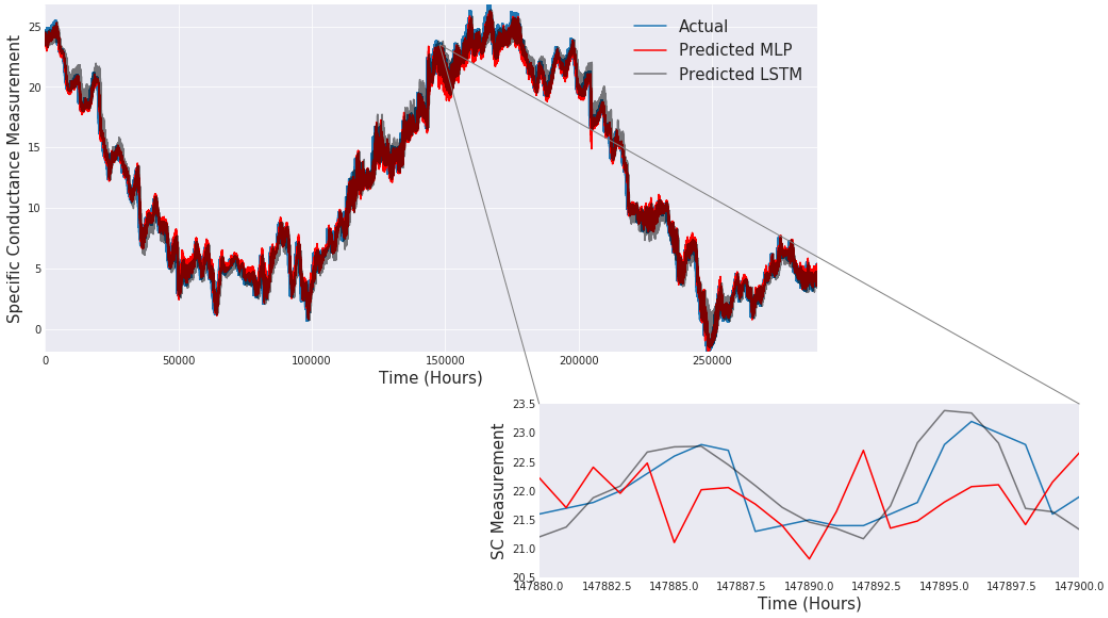


Figure 5.3: Day-to-day temperature prediction, using 24 hour values.

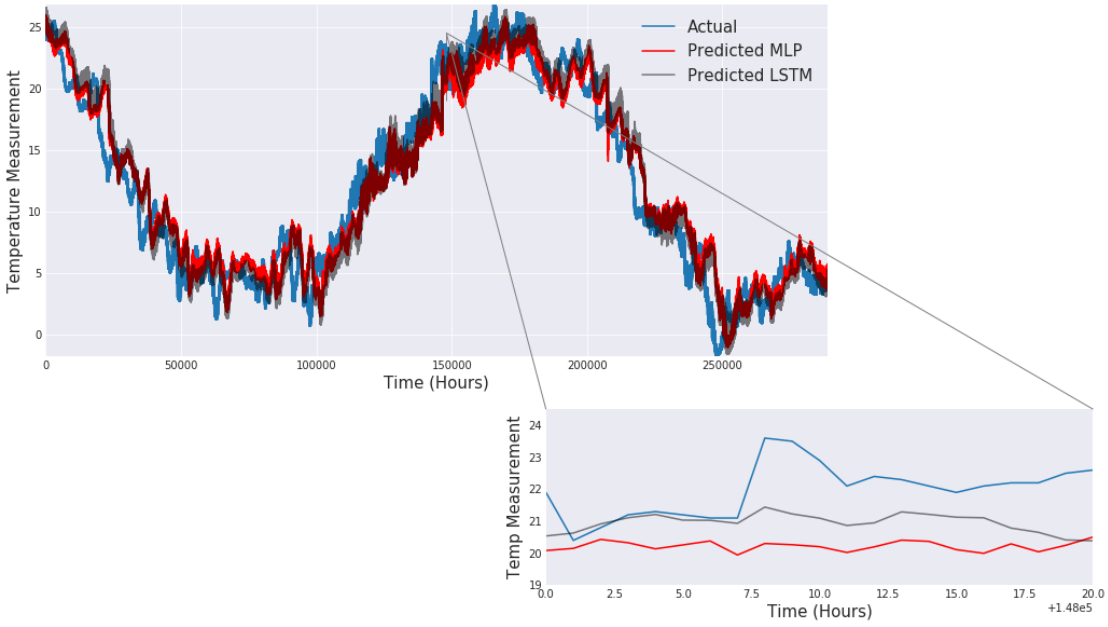


Figure 5.4: Temperature prediction one week ahead, using 24 hour values.

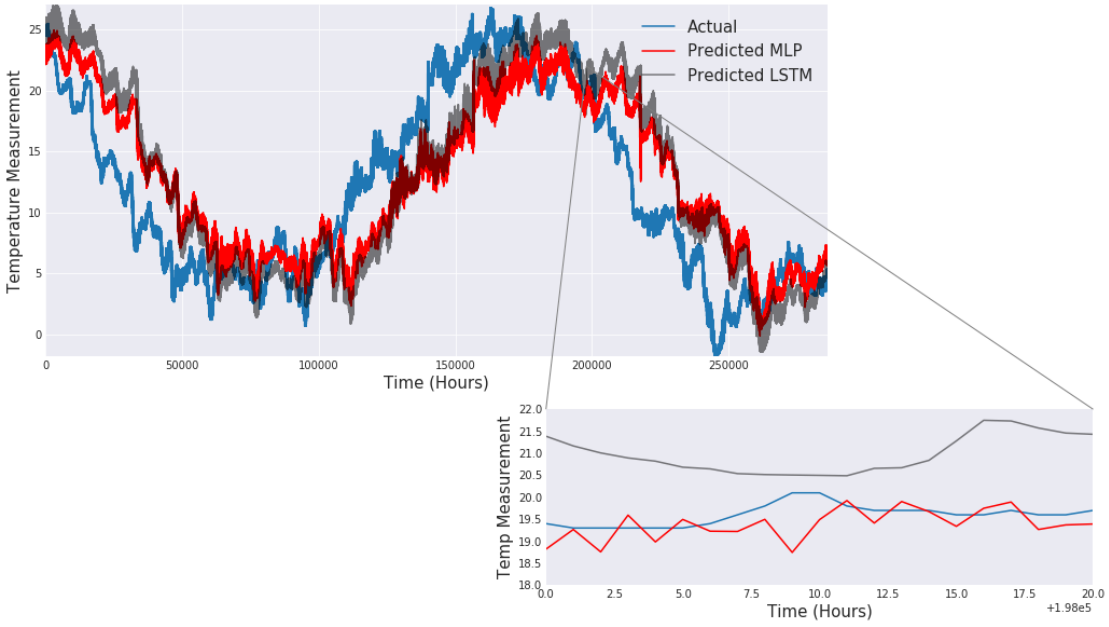


Figure 5.5: Temperature prediction one month ahead, using 24 hour values.

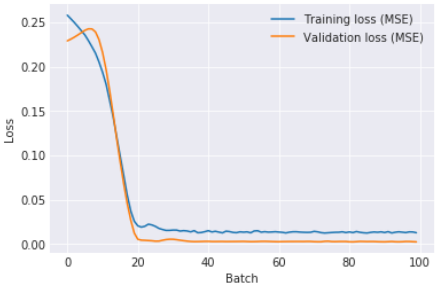
the next day's temperature. When 24 hourly temperature measurements are considered in the prediction of the next day's 24 hourly temperatures, LSTM is closer to the actual values than MLP, as can be seen in Figure 5.3. In the case of predicting the 24 hourly temperature values in the following week, the prediction ability of both models drops. The prediction ability decreases even more when predicting 24 hourly temperature values one month ahead. Figures 5.4 and 5.5 present the predicted temperature values for these two cases. In both cases, LSTM predicted values closer to the actual values than MLP.

5.3 Training and Validation Loss

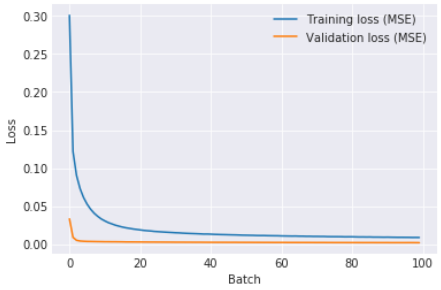
Mean squared error (MSE) is the loss function used in measuring the fitness of MLP and LSTM on the dataset, during training. Figure 5.6 shows the training and validation MSE to the number of training epochs. As mentioned in Section 5.1, 100 epochs were used to train both models.

Figure 5.6a shows that the validation loss for MLP is lower than the training loss; which indicates that the values in the validation dataset might be easier to predict compared to the training dataset. However, there is a noticeable gap between the two sets. In Figure 5.6b the training and validation curves show improvement after the 20th epoch, possibly because of the small sample of the

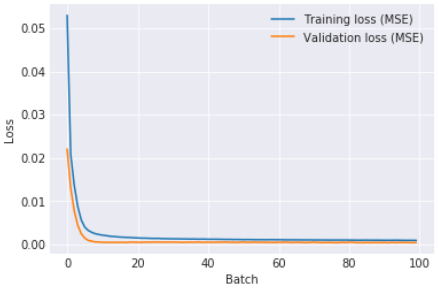
(a) MLP: 1to1 day prediction.



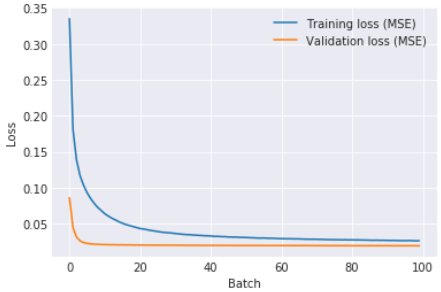
(b) LSTM: 1to1 day prediction.



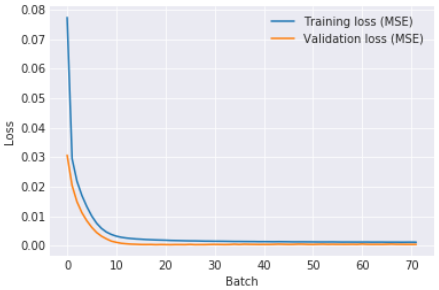
(c) MLP: 24to24 day prediction.



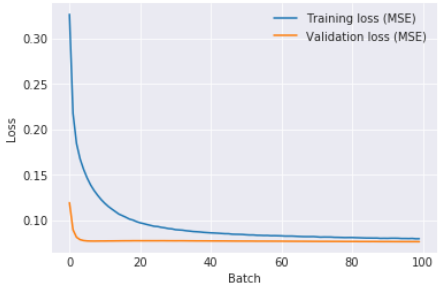
(d) LSTM: 24to24 day prediction.



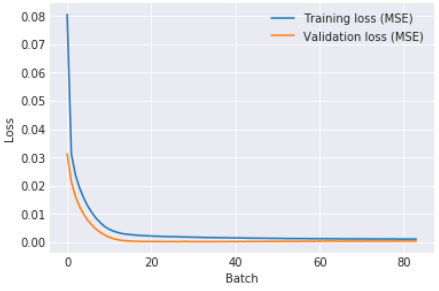
(e) MLP: 24to24 week prediction.



(f) LSTM: 24to24 week prediction.



(g) MLP: 24to24 month prediction.



(h) LSTM: 24to24 month prediction.

Figure 5.6: Training and validation loss (MSE) during training of the various models, for the prediction of temperature.

observations used in this experiment. The rest of the figures have minimal gaps between the final loss values. Figures 5.6f and 5.6h demonstrate the loss values over fewer epochs, due to the regularisation technique of early stoppage (refer to Section 3.1.7).

5.4 Accuracies of the Models

After performing prediction on the test set, the predicted values were transformed to their original ranges using the inverse normalisation function. The accuracy measures described in Section 3.5 (namely mean squared error, mean absolute error, median absolute error, R^2 and mean absolute percentage error) were computed to evaluate the performance of the two models. The results of the four experiments are reported. Table 5.2 shows the accuracy measures for the day-to-day predictions, where one hour per day was used as an input to MLP and LSTM. Table 5.3 presents the accuracy measures for the day-to-day predictions where 24 hourly values per day were used as input to the models. Table 5.4 gives the accuracy measures for the prediction of 24 hourly values one week ahead, where 24 hourly values were used as input. Table 5.5 shows the accuracy measures for the prediction of 24 hourly values one month ahead, where 24 hourly values were used as input.

Tables 5.2, 5.3, 5.4 and 5.5 indicate that the prediction of specific conductance resulted in the worst accuracy for both MLP and LSTM models, according to the MSE, MAE and MedAE values. In contrast, the prediction of pH is significantly more accurate for both MLP and LSTM models in terms of MSE, MAE as well as MedAE.

Table 5.2 illustrates that temperature is predicted with achieves the highest accuracy according to the R^2 measure, for both MLP and LSTM. Table 5.2 also shows that the difference between observed and predicted elevation variable leads to the lowest accuracy compared to the rest of the variables using the R^2 and MAPE measures. The pH variable, on the other hand, achieves the highest accuracy in both MLP and LSTM models using the MAPE measure.

All the experiments that were performed in this study can be found on GitHub repositories[2].

[2] https://github.com/ReemOmer/WQV_Exploratory_Data_Analysis, https://github.com/ReemOmer/WQV_Anomaly_Detection, https://github.com/ReemOmer/WQV_Transfer_Learning, https://github.com/ReemOmer/WQV_Predictive_Models

5.5 Discussion

Looking at Tables 5.2, 5.3, 5.4 and 5.5, it is observed that specific conductance was not accurately predicted by the MLP and LSTM models. Although the percentage of missing values in the specific conductance time-series was not high (9.98%, as seen in Section 4.5), the accuracy of predicting different times with several inputs (for example one hour to one hour and 24 hours to 24 hours) was low compared to the rest of the variables in the dataset. A possible reason could be the high fluctuation rates present in the specific conductance series.

Generally, predicted pH values were closest to the actual values for all four experiments. This can be due to the data availability as well as the validation and testing data being similar to the training data.

It is also found that MLP uses the 100 epochs in all the experiments while LSTM uses 40, 60, 80 or 100 epochs. Also, the MLP experiments had a short training time for all the variables (26.95 minutes on average), while LSTM required 44 hours to train over one variable. Practically, the LSTM models achieved higher accuracy compared to the MLP models for most of the accuracy measures. Figure 5.6 shows that LSTM models were also able to converge in a smaller number of training epochs compared to MLP.

Finally, LSTM was able to capture the dynamics of the water quality variables better than MLP. This might be due to the nature of the data being sequential, and LSTM being an artificial neural network designed specifically to deal with sequential data. The memory cell in the LSTM allows it to save the fluctuations of the series for future predictions.

Compared to the study by Khan and See [35], they use only one artificial neural network model to predict some of the water quality variables of Hog Island, while this study uses two models: MLP and LSTM. Khan and See considered four series: chlorophyll, specific conductance, dissolved oxygen and turbidity, whereas this study applied the models to twelve variables (including the four mentioned above). Khan and See used only the MSE to present the accuracy, and this study considered five difference accuracy measures.

5.6 Summary

This chapter presented the architecture of the MLP and LSTM models in Section 5.1, which were used in performing the predictions. The results were presented in Section 5.2, and the losses that were obtained from training MLP and LSTM were discussed in Section 5.3. Moreover, the accuracy of the models were com-

pared using MSE, MAE, MedAE, R^2 and MAPE in Section 5.4. Lastly, the obtained results were discussed in Section 5.5. The next chapter will conclude the study and suggest possible improvements.

Table 5.2: Performance metrics on the test set, of the day-to-day models using one hour per day.

Variable	MSE		MAE		MedAE		R^2		MAPE	
	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM
Specific conductance	549959.79	394689.01	579.65	452.11	504.16	334.13	0.82	0.87	1.25	0.99
Dissolved oxygen	0.42	0.42	0.4	0.41	0.26	0.26	0.92	0.92	4.58	4.63
Chlorophyll	35.34	62.07	4.13	5.25	2.78	3.45	0.84	0.73	28.34	38.2
Turbidity	2.34	2.45	0.94	1.03	0.67	0.8	0.23	0.2	28.51	32.69
Temperature	0.62	0.71	0.58	0.62	0.46	0.47	0.99	0.99	11.19	14.17
Sampling depth	1.89	1.66	1.1	1.02	0.99	0.78	0.68	0.72	20.66	19.18
pH	0.01	0.01	0.05	0.05	0.02	0.02	0.9	0.9	0.6	0.62
Chlorophylls	89.95	167.99	5.54	7.92	3.06	4.94	0.79	0.61	36.8	58.55
Surface elevation	0.92	0.93	0.75	0.76	0.63	0.63	0.69	0.69	124.62	122.94
Tidal prediction	0.31	0.32	0.48	0.48	0.43	0.46	0.85	0.84	118.69	109.14
Salinity	0.38	0.21	0.49	0.34	0.44	0.25	0.76	0.86	1.63	1.14
Difference obs-prd elevation	1.57	2.15	0.99	1.18	0.92	1.03	0.09	-0.25	128.27	246.77

Table 5.3: Performance metrics on the test set, of the day-to-day models using 24 hourly values per day.

Variable	MSE		MAE		MedAE		R^2		MAPE	
	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM
Specific conductance	406029.78	5842.16	458.22	43.97	345.14	41.32	0.86	1.0	1.0	0.09
Dissolved oxygen	0.48	0.05	0.43	0.17	0.28	0.13	0.91	0.99	4.88	2.18
Chlorophyll	34.71	0.36	4.19	0.37	2.97	0.22	0.83	1.0	34.85	2.61
Turbidity	8.61	1.83	1.34	0.43	0.97	0.19	-0.16	0.75	43.54	10.3
Temperature	0.53	0.04	0.53	0.11	0.4	0.07	0.99	1.0	10.46	2.93
Sampling depth	1.99	0.02	1.12	0.11	0.93	0.07	0.45	0.99	18.82	1.67
pH	0.01	0.0	0.05	0.0	0.04	0.0	0.9	1.0	0.64	0.06
Chlorophylls	105.81	2.16	6.12	0.73	3.44	0.51	0.7	0.99	46.19	6.36
Surface elevation	0.54	0.01	0.53	0.06	0.39	0.04	0.82	1.0	105.07	10.46
Tidal prediction	0.18	0.02	0.32	0.09	0.26	0.07	0.91	0.99	48.78	17.75
Salinity	0.22	0.0	0.34	0.02	0.26	0.02	0.85	1.0	1.14	0.08
Difference obs-prd elevation	1.11	0.02	0.82	0.08	0.64	0.05	0.15	0.98	181.52	19.75

Table 5.4: Performance metrics on the test set, of the one-week-ahead models using 24 hourly values per day.

Variable	MSE		MAE		MedAE		R^2		MAPE	
	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM
Specific conductance	1402212.4	7344.72	929.06	63.97	777.65	50.75	0.53	1.0	2.01	0.14
Dissolved oxygen	1.97	0.01	0.97	0.05	0.67	0.04	0.64	1.0	10.81	0.53
Chlorophyll	101.96	1.83	7.59	0.77	5.59	0.52	0.49	0.99	70.78	6.36
Turbidity	11.28	5.59	1.8	0.92	1.38	0.54	-0.52	0.24	60.69	23.67
Temperature	4.81	0.05	1.73	0.14	1.42	0.08	0.92	1.0	40.27	3.3
Sampling depth	3.43	0.04	1.48	0.14	1.25	0.09	0.05	0.99	25.01	2.05
pH	0.03	0.0	0.13	0.01	0.09	0.0	0.42	1.0	1.63	0.07
Chlorophylls	197.97	6.37	9.12	1.15	5.44	0.56	0.44	0.98	76.82	7.46
Surface elevation	1.97	0.01	1.09	0.05	0.87	0.04	0.36	1.0	211.22	8.56
Tidal prediction	1.12	0.05	0.87	0.13	0.76	0.08	0.4	0.98	124.29	24.43
Salinity	0.73	0.0	0.67	0.03	0.56	0.02	0.52	1.0	2.23	0.09
Difference obs-prd elevation	1.87	0.21	1.11	0.3	0.95	0.2	-0.55	0.79	316.25	42.52

Table 5.5: Performance metrics on the test set, of the one-month-ahead models using 24 hourly values per day.

Variable	MSE		MAE		MedAE		R^2		MAPE	
	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM
Specific conductance	2785755.7	7244.36	1344.21	66.15	1175.25	67.59	0.07	1.0	2.89	0.14
Dissolved oxygen	3.59	0.01	1.44	0.05	1.04	0.04	0.33	1.0	15.97	0.59
Chlorophyll	282.65	1.41	13.35	0.78	10.61	0.52	-0.38	0.99	129.5	4.46
Turbidity	11.27	4.17	2.09	1.01	1.71	0.65	-0.51	0.2	74.97	24.89
Temperature	20.6	0.06	3.77	0.13	3.58	0.07	0.64	1.0	83.67	3.15
Sampling depth	3.55	0.08	1.52	0.2	1.3	0.14	-0.35	0.96	30.38	3.4
pH	0.05	0.0	0.16	0.0	0.13	0.0	0.13	1.0	2.1	0.05
Chlorophylls	344.93	2.06	12.88	0.78	8.94	0.52	0.04	0.99	120.77	6.96
Surface elevation	3.66	0.0	1.56	0.05	1.37	0.04	-0.19	1.0	325.38	8.3
Tidal prediction	2.17	0.14	1.24	0.23	1.16	0.12	-0.49	0.91	200.23	54.28
Salinity	1.43	0.01	0.96	0.09	0.84	0.08	0.06	0.99	3.19	0.31
Difference obs-prd elevation	0.68	0.06	0.65	0.17	0.52	0.13	-0.17	0.84	324.67	96.76

Chapter 6

Conclusion and Future Work

This chapter is divided into two sections. Section 6.1 concludes the results and findings of the study, and Section 6.2 presents possible extensions to the study.

6.1 Conclusion

In this study, multilayer perceptron (MLP) and long short-term memory (LSTM) models were implemented to predict the water quality variables recorded at Hog Island in the United States. The models were trained on a dataset obtained from the United States Geological Survey, that contains twelve variables with different scales as shown in Section 4.2.

The scheme that was used to obtain the results was the following:

- data preprocessing techniques were performed, including
  - testing and converting non-stationary time-series into stationary time-series (as discussed in Section 4.4);
  - infilling gaps in the time-series using the rolling window and the adjacent mean technique (as indicated in Section 4.5);
  - detecting and replacing outliers with an ensemble of various techniques (as shown in Section 4.7.2);
  - normalising the scale of the water quality variables (as demonstrated in Section 4.8);
- data splitting into training, validation and testing sets (as explained in Section 4.9);

- training the models using optimal hyperparameters on the preprocessed, scaled and split datasets (refer to Section 5.1);
- prediction of future values of water quality variables were performed and inversed back to their original scales;
- the results of MLP and LSTM were compared using the mean squared error, mean absolute error, median absolute error, mean absolute percentage error and R^2 score metrics.

The results of the comparison showed that LSTM achieved higher accuracy using several accuracy metrics in predicting hourly values of water quality variables, while MLP achieved lower accuracies. Therefore, it may well be argued that the LSTM model fitted the data better than the MLP model. Although the models gave adequate results, some variables did not have a sufficient amount of data for the training process, which resulted in low prediction accuracies.

6.2 Future Work

There are a number of possible extensions that could be added to this work. Some are listed below.

- The MLP and LSTM models can be applied to different water quality datasets, and a good model can be chosen for the prediction of water quality variables in general.
- Some variables have strong correlation with others. It may be useful to study if a variable with more data is used to predict other variables with fewer observations.
- An advanced system could be built by adding a classification model to define the usage of the water, based on the predicted results. This could be done after comparing the predicted values from MLP and LSTM with thresholds for the designated use.
- In addition, different machine learning models can be experimented with, such as gated recurrent units and one-dimensional convolutional neural networks.
- The models can be applied to other environmental problems like weather prediction, by transferring the knowledge gained from this study.

List of References

[1] Components of a time series. https://cmapskm.ihmc.us/rid=1052458821502_1749267941_6906/components.pdf. Online; accessed 19 November 2019.

[2] Kmeans clustering technique. https://www.kdnuggets.com/2016/09/comparing-clustering-techniques-concise-technical-overview.html. Online; accessed 30 May 2019.

[3] Nitrogen and water quality. http://www.state.ky.us/nrepc/water/ramp/rmnox.htm. Online; accessed 21 March 2019.

[4] Non stationary time series Python. https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/. Online; accessed 22 May 2019.

[5] pH in the environment ecosystem. https://www.water-research.net/index.php/ph-in-the-environment. Online; accessed 21 February 2018.

[6] Study session 3 water sources and their characteristics. https://www.open.edu/openlearncreate/pluginfile.php/168317/mod_oucontent/oucontent_download/printable/dbc20d332a97439a440439b6788fa40fde54fc36/study_session_3__water_sources_and_their_characteristics_printable.pdf?downloaded=1&timestamp=1576023811048. Online; accessed April 2018.

[7] Unit root test. https://en.wikipedia.org/wiki/Unit_root_test. Online; accessed 22 May 2019.

[8] Ratnadip Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting. *CoRR*, abs/1302.6613, 2013.

[9] Shivani Agarwal. Lecture notes machine learning, online learning (and perceptron), April 2019. University of Pennsylvania.

[10] Daniel Kelly Christine Kemker Kevin Rose Alex Card, Katie Fitch. Environmental measurements. https://www.fondriest.com/environmental-measurements/. Online; accessed October 2018.

[11] Afshine Amidi and Shervine Amidi. Recurrent neural networks cheatsheet. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. Online; accessed October 2019.

[12] Wenjuan An, Mangui Liang, and He Liu. An improved one-class support vector machine classifier for outlier detection. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 229(3):580–588, 2015.

[13] Pius Adebayo Araoye. The seasonal variation of pH and dissolved oxygen (DO_2) concentration in Asa Lake Ilorin, Nigeria. *International Journal of Physical Sciences*, 4, May 2009.

[14] Tirthankar Banerjee and Dr. R. Srivastava. Application of water quality index for assessment of surface water quality surrounding integrated industrial estate-pantnagar. *Water science and technology : a journal of the International Association on Water Pollution Research*, 60:2041–53, October 2009.

[15] Russell Beale and Tom Jackson. *Neural Computing: An Introduction*. IOP Publishing Ltd., Bristol, UK, 1990.

[16] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[17] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. volume 105, pages 17–24, November 2014.

[18] Anthony L. Caterini. A novel mathematical framework for the analysis of neural networks. Master's thesis, Applied Mathematics, University of Waterloo, August 2017.

[19] Wen-Huan Chine, Tai-Sheng Wang, Li Chen, and Chang-Huan Kou. Artificial neural networks for water quality prediction in a reservoir. In *2009 Second International Workshop on Computer Science and Engineering*, volume 1, pages 516–519. IEEE, 2009.

[20] Chuah. Water quality study of the east johor strait. Master's thesis, Department of Chemical Engineering, University of Singapore, 1998.

[21] EluÃ£ Ramos Coutinho, Robson Mariano Da Silva, Jonni Guiller Ferreira Madeira, Pollyanna Rodrigues de Oliveira Dos Santos Coutinho, Ronney Arismel Mancebo Boloy, and Angel Ramon Sanchez Delgado. Application of artificial neural networks (ANNs) in the gap filling of meteorological time series. *Revista Brasileira de Meteorologia*, 33(2):317–328, 2018.

[22] S Holmes CSIR Environmental Services. South African water quality guidelines. agricultural use: Irrigation, 1996.

[23] S Holmes CSIR Environmental Services. South African water quality guidelines. domestic use, 1996.

[24] S Holmes CSIR Environmental Services. South African water quality guidelines. industrial use, 1996.

[25] Kwetishe Joro Danjuma. Performance evaluation of machine learning algorithms in post-operative life expectancy in the lung cancer patients. *CoRR*, abs/1504.04646, 2015.

[26] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATSâ10). Society for Artificial Intelligence and Statistics*, 2010.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[28] S. Harzallah, R. Rebhi, M. Chabaat, and A. Rabehi. Eddy current modelling using multi-layer perceptron neural networks for detecting surface cracks. *Frattura ed Integrita Strutturale*, 12(45):147–155, 2018.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[30] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5:01–11, March 2015.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[32] Aarshay Jain. Ts forecasting python. https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/. Online; accessed 22 May 2019.

[33] Skipper Seabold Josef Perktold and Jonathan Taylor. statsmodels adfuller. https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html. Online; accessed 21 May 2019.

[34] Serena Yeung Justin Johnson and Fei-Fei Li. Lecture notes in convolutional neural networks for visual recognition, September 2019. Stanford University.

[35] Yafra Khan and Chai Soo See. Predicting and analyzing water quality using machine learning: A comprehensive model. In *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–6. IEEE, 2016.

[36] Sofiane Khelifa, Bachir Gourine, Habib Taibi, and Hicham Dekkiche. Filling gaps in time series of space-geodetic positioning. *Arabian Journal of Geosciences*, 11(12):1–7, 2018.

[37] Sungil Kim and Heeyoung Kim. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32:669–679, 07 2016.

[38] Richard Klein. The human career: Human biological and cultural origins. *Evolution*, 45, January 2009.

[39] Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *NIPS*, 1991.

[40] Jinhao Lei, Chao Liu, and Dongxiang Jiang. Fault diagnosis of wind turbine based on long short-term memory networks. *Renewable Energy*, 133:422–432, 2019.

[41] Chuanqi Li and Wei Wang. Assessment of the water quality near the dam area of three gorges reservoir based on bayes. *Information Science and Engineering, International Conference on*, 0:145–148, January 2009.

[42] Yidong Liu, Siting Liu, Yanzhi Wang, Fabrizio Lombardi, and Jie Han. A stochastic computational multi-layer perceptron with backward propagation. *IEEE Transactions on Computers*, 67(9):1273–1286, 2018.

[43] Eelco Loucks, Daniel P.and van Beek. *Water Quality Modeling and Prediction*, pages 417–467. Springer International Publishing, Cham, 2017.

[44] Jinsuo Lu and Tinglin Huang. Data mining on forecast raw water quality from online monitoring station based on decision-making tree. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 706–709. IEEE, 2009.

[45] Puggini Luca and Mcloone Sean. An enhanced variable selection and isolation forest based methodology for anomaly detection with oes data. *Engineering Applications of Artificial Intelligence*, 67:126–135, 2018.

[46] Salisu Yusuf Muhammad, Mokhairi Makhtar, Azilawati Rozaimee, Azwa Abdul Aziz, and Azrul Amri Jamal. Classification model for water quality using machine learning techniques. *International Journal of Software Engineering and Its Applications*, 9(6):45–52, 2015.

[47] Pushparaja Murugan. Learning the sequential temporal information with recurrent neural networks. *CoRR*, abs/1807.02857, 2018.

[48] Zuriani Mustaffa and Yuhanis Yusof. A comparison of normalization techniques in predicting dengue outbreak. 2010.

[49] Michael Negnevitsky. *Artificial intelligence: a guide to intelligent systems*. Pearson Education, 2005.

[50] Andrew Ng. Lecture notes in machine leanring. https://www.coursera.org/learn/machine-learning/supplement/d5Pt1/lecture-slides, 2017.

[51] Ismoilov Nusrat and Sung-Bong Jang. A comparison of regularization techniques in deep neural networks. *Symmetry*, 10:648, November 2018.

[52] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *CoRR*, abs/1811.03378, 2018.

[53] World Health Organization. *Guidelines for drinking-water quality: fourth edition incorporating first addendum*. World Health Organization, 4th ed + 1st add edition, 2017.

[54] Sundarambal Palani, Shie-Yui Liong, and Pavel Tkalich. An ANN application for water quality forecasting. *Marine Pollution Bulletin*, 56(9):1586–1597, 2008.

[55] Chadaphim Photphanloet, Weeris Treeratanajaru, Nagul Cooharojananone, and Rajalida Lipikorn. Biochemical oxygen demand prediction for Chaophraya River using alpha-trimmed ARIMA model. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6. IEEE, 2016.

[56] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 55–69, London, UK, UK, 1998. Springer-Verlag.

[57] Vishal R Y V Lokeswari Raghav Nandakumar, Uttamraj K R. Stock price prediction using long short term memory. *International Research Journal of Engineering and Technology(IRJET)*, 05, March 2018.

[58] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017.

[59] Zuzana Reitermanova. Data splitting. pages 31–36, Prague, Czech Republic, 2010. Charles University. ISBN 9788073781392.

[60] Dina A. Salem, Rania A. Abul Seoud, and Yasser M. Kadah. Prediction of binding peptides to class i major histocompatibility complex using modified scoring matrices and data splitting strategies. *Biocybernetics and Biomedical Engineering*, 36(3):509 – 520, 2016.

[61] Bernhard Scholkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. *Support Vector Method for Novelty Detection*. Number 0-262-11245-0. MIT Press, 1998.

[62] Luai Shalabi, Shaaban Zyad, and Basil Al-Kasasbeh. Data mining: A preprocessing engine. *Journal of Computer Science*, 2, September 2006.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[64] Sourabh Suman, B. Rajathilagam, and Karthik Vaidhyanathan. A generic approach for outlier detection in time-series data. *Indian Journal of Scientific Research*, 2017.

[65] Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[66] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[67] Zhenxiang Xing, Qiang Fu, and Dong Liu. Water quality evaluation by the fuzzy comprehensive evaluation based on ew method. In *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 1, pages 476–479. IEEE, 2011.

[68] Yan Yan, Ying Wang, Wen-Chao Gao, Bo-Wen Zhang, Chun Yang, and Xu-Cheng Yin. Lstm2: Multi-label ranking for document classification. *Neural Processing Letters*, 47(1):117–138, February 2018.

[69] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521, May 2013.

[70] Changjun Zhu and Zhenchun Hao. Fuzzy neural network model and its application in water quality evaluation. In *2009 International Conference on Environmental Science and Information Application Technology*, volume 1, pages 251–254. IEEE, 2009.